

Forward Thinking in Reverse

*Design, Implementation, and Continuous Monitoring of a Closed-Loop Supply Chain
Using Optimization, Simulation, and Dashboard Systems to Maximize Net Recovery*

By Timothy J. Vasil

A.B. Computer Science, Harvard University, 2003

Submitted to the MIT Sloan School of Management and the Department of Electrical Engineering and
Computer Science in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Electrical Engineering and Computer Science

in conjunction with the Leaders for Global Operations Program at the
Massachusetts Institute of Technology

June 2011

© 2011 Timothy J. Vasil. All rights reserved.

ARCHIVES

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUN 15 2011

LIBRARIES

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic
copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of author



Department of Electrical Engineering and Computer Science
MIT Sloan School of Management
May 6, 2011

Certified by



Abbott Weiss, Thesis Supervisor
Senior Lecturer, Department of Mechanical Engineering

Certified by

s H. Fine, Thesis Supervisor
Chrysler LGO Professor of Management and Engineering Systems

Accepted by

Leslie Kolodziejski, Professor of Electrical Engineering and Computer Science
Chair, EECS Committee on Graduate Students

Accepted by

Debbie Berechman, Executive Director of MBA Program
MIT Sloan School of Management

This page intentionally left blank.

Forward Thinking in Reverse

*Design, Implementation, and Continuous Monitoring of a Closed-Loop Supply Chain
Using Optimization, Simulation, and Dashboard Systems to Maximize Net Recovery*

By Timothy J. Vasil

Submitted to the MIT Sloan School of Management and the Department of Electrical Engineering and Computer Science on May 6, 2011 in partial fulfillment of the requirements for the degrees of Master of Business Administration and Master of Science in Electrical Engineering and Computer Science

Abstract

Developed during our recent six-month engagement at Dell—a leading computer manufacturer and services provider with one of the world’s leading supply chains—we discuss a network flow-based mixed-integer linear program (MILP) model to identify the critical factors in optimizing reverse supply chain design decisions to optimize profit. The model is fast, intuitive, flexible, and robust to uncertainty. Its outputs include specific design recommendations, financial impact estimates, dynamically generated product routing diagrams, and multi-scenario sensitivity analysis. Through two case studies, the first in U.S. smartphone returns and the second in Europe, Middle East and Africa (EMEA) Alienware-branded computer returns, we show how our model fosters standardized, robust strategic decision-making and serves as a platform upon which to build management systems for continuous improvement. We then discuss two such systems: a simulation-based reusable packaging cost-benefit analysis (CBA) calculator, and an automated dashboard for managing disassembly-for-parts decisions.

Thesis Supervisor: Abbott Weiss

Title: Senior Lecturer, Department of Mechanical Engineering

Thesis Supervisor: Charles H. Fine

Title: Chrysler LGO Professor of Management and Engineering Systems

This page intentionally left blank.

Acknowledgements

Throughout my six-month internship at Dell, I enjoyed collaborating with many approachable and insightful employees, all of whom have been instrumental in shaping my work. Bryan Hood (program manager) oversaw my project and provided me with exceptional support. Matt Snyder (global reverse logistics manager) offered clear guidance, expert insight, and unwavering dedication. Anju Jaggi served as my mentor and ensured I had the resources I needed while at Dell. Annette Clayton (VP, supply chain) sponsored my internship and supports the MIT Leaders for Global Operations (LGO) program. Participants of the global, cross-functional working group—Amy Bao, Michelle Butler, Grace Gao, David Giese, Conception Kirchoff, Mark Landrum, Pat Lillis, Aizaz Manzar, Ken Miller, Patricia Ring, Connie Quintanilla, JR Rogers, and David Vignini—shared their real-world expertise, offered constructive feedback, contributed to the refinement of the asset recovery model presented herein, and enthusiastically participated in case studies. Reusable packaging advocates Oliver Campbell, Ed Escamilla, Elena Espinel, and Matt Sorenson aided me in performing the appropriate financial calculations for cost/benefit analysis. Dashboard collaborations Vishal Balchandani, Kevin Myers, John Schiefelbein, and Vipin Varkey dedicated many hours of their time to get me up to speed on the mechanics of teardown decision making.

I am especially indebted to Jack DeButts (program manager), who took a keen interest in my success, providing me with not only an extraordinary introduction to Dell's reverse supply chain, but reverse logistics in general. Together we toured return centers, met with numerous third-party logistics providers (3PLs), a retail return center, and engaged with the Reverse Logistics Association (RLA). He also introduced me to some of the best barbeque in central Texas!

From MIT, my thesis advisors, Abbott Weiss and Charlie Fine, provided timely guidance. I thank them for their time and unique perspectives. I also wish to thank David Segrera, Steve Wessels, and Leo Espindle, three of my classmates, for their help with sensitivity analysis, dynamic chart generation, and n-gram analysis, respectively.

Finally, I wish to acknowledge Alex and my family for their invaluable support and advice. They encouraged me to pursue a graduate education and have been by my side throughout the process.

This page intentionally left blank.

Table of Contents

Abstract.....	3
Acknowledgements	5
Table of Contents.....	7
Table of Topics.....	9
Table of Figures	11
Table of Tables.....	15
Table of Listings.....	17
Chapter 1. Introduction.....	19
1.1. Overview of Dell.....	19
1.1. Problem Statement	22
1.2. Hypothesis.....	24
1.3. Research Methodology	24
1.4. Thesis Outline	25
Chapter 2. Literature Review	27
2.1. Closed-loop Supply Chain Management Overview	28
2.2. Surveys & Trends.....	37
2.3. Diagnostic Tools.....	39
2.4. System Dynamics Simulation Models.....	39
2.5. Discrete Event Simulation Models.....	40
2.6. Economic Models.....	41
2.7. Optimization Models.....	42
2.8. Key Distinctions between This Research and Prior Publications	47
Chapter 3. Organizational Assessment.....	49
3.1. Strategy.....	49
3.2. Three Lens Analysis.....	50
3.3. Reverse Supply Chain Maturity Assessment	56
3.4. Driving Change: Techniques for Success at Dell	65
Chapter 4. Optimization System for Supply Chain Design.....	69
4.1. Overview	70
4.2. Model Formulation.....	74
4.3. Implementation Overview	85
4.4. User Inputs.....	90
4.5. Recommendation Engine.....	100
4.6. Outputs.....	106
4.7. Summary.....	114
Chapter 5. Small Form Factor Returns Case Study	116
5.1. Overview	116
5.2. Baseline Scenario Estimation	117
5.3. Sensitivity Analysis	121
5.4. Capability Recommendations.....	126
5.5. Generalizable Insights	126

Chapter 6. Alienware Repair Case Study.....	128
6.1. Overview	128
6.2. Calibration	129
6.3. Impact Analysis.....	131
6.4. Sensitivity Analysis.....	133
6.5. Generalizable Insights	138
Chapter 7. Reusable Packaging Case Study	139
7.1. Overview	139
7.2. Implementation	140
7.3. Sourcing of Packaging Options.....	143
7.4. Evaluation	144
7.5. Generalizable Insights	147
Chapter 8. Teardown Dashboard Case Study	149
8.1. Overview	149
8.2. Dashboard Design	151
8.3. Data Aggregation	155
8.4. User Interface Implementation.....	157
8.5. Generalizable Insights	161
Chapter 9. Conclusions and Next Steps.....	163
9.1. Specific Recommendations to Dell.....	164
9.2. General Implications	166
9.3. Remaining Research Questions.....	167
Appendix A. Glossary	169
A.1. Dell-specific Terminology	169
A.2. ARM Model Terminology.....	170
A.3. Closed-loop Supply Chain Terminology	171
Appendix B. ARM Model Source Code.....	180
B.1. Network Flow Diagram Drawing.....	180
B.2. Benefit-per-box Calculations	181
B.3. Sensitivity Analysis	184
Appendix C. Reusable Packaging CBA Source Code	186
C.1. Discrete Event Simulator	186
C.2. CBA Spreadsheet Integration.....	188
Appendix D. Teardown Dashboard Source Code.....	190
D.1. Connections Table.....	190
D.2. Control Panel	191
References.....	197

Table of Topics

We endeavor to cover many topics in this paper—including reverse supply chain design, mathematical modeling and sensitivity analysis, data visualization, and reusable packaging. Anticipating most readers will be interested in only a subset of these topics, we offer the topic-based guide below so you can quickly find information pertinent to your specific areas of interest.

Reverse Supply Chain Design

Background information on the intent, capabilities, and potential of the reverse supply chain

Reverse supply chain design questions	21
Strategic reasons for handling returns	37
Diagnostic tools	39
Optimal reverse supply chain designs	41
Reverse supply chain functions at Dell	52
General implications	166
EVP of Reverse Logistics role	167
Terminology	171

Dell's Supply Chain

An assessment of Dell's current closed-loop supply chain capabilities and opportunities for improvement

Three lens analysis	50
3PL partners	51
Collection methods	58
Capabilities	61
GENCO case study	63
SFF returns	117
Alienware repair in EMEA	131
Teardown activities	152
Recommendations	164
Terminology	169

Sustainability

Green initiatives in the context of the closed-loop supply chain

Eco-centric handling of product returns	28
Reusable packaing options	143
Limitations of commonly used reusable packaging calculators	147
Product design considerations	165

Metrics

Critiques of the measurements and incentives used in managing reverse supply chain capabilities

Detailed description of the “net recovery” metric	22
Key reverse supply chain metrics	71
Teardown metrics	159
Visualizing metrics as part of a continuous improvement process	161
Accounting techniques	53
Asset Recovery Maximizer (ARM) model:	
Differences from existing models	47
Profit-maximizing objective function	81
Financial summary	108

Models

Mathematical models for reverse supply chain decision-making from both our research and prior work

Prior work (literature review):	
System dynamics models	39
Discrete event simulation models	40
Optimization models	42
Asset Recovery Maximizer (ARM) model:	
Definition	74
Terminology	170
Source code	180
Reusable packaging simulation model:	
Definition	140
Source code	186
Sensitivity analysis:	
ARM model capabilities	111
Small form factor case study	121
Alienware repair case study	133
Reusable packaging case study	146

Data Visualization

How to collect, aggregate, and present data in a useful and intuitive way

Network flow diagram	111
Teardown dashboard:	
Design	151
Implementation	140
Insights	161
Source code	190

Smartphones & Tablets

A description of Dell's SFF products, including supporting business and operations structures

Product line	20
Closed-loop supply chain planning	54
Capabilities assessment	61
Case study	116

Leading Initiatives at Dell

Techniques for managing teams, initiating improvements, and conducting case studies

Strategic fit	49
Techniques for driving change	65
Our engagement:	
Phases and timeline	24
Context within Dell's strategic pillars	50
User's Guide deliverable	68
Case study methodologies:	
Small form factor returns	116
Alienware repair in EMEA	129
Reusable packaging	140
Teardown dashboard	150

Table of Figures

Figure 1:	Dell's SFF product offerings to date.....	20
Figure 2:	Apple revenue over the last 6 years by product line [1]	20
Figure 3:	Visual explanation of the <i>net recovery</i> metric. While <i>recovery revenue</i> often is less than <i>COGS</i> and original <i>revenue</i> (as shown), this need not be the case.....	22
Figure 4:	Phases and timeline of our six-month engagement with Dell.....	24
Figure 5:	The three types of supply chains [5].....	27
Figure 6:	Lansink's ladder depicting a eco-centric hierarchy of handling product returns [12]	28
Figure 7:	Reverse logistics process described by Kumar and Putnam [13]	30
Figure 8:	Effect of reverse logistics on profitability across various return policies	34
Figure 9:	Aspects of reverse logistics from Dowlatshahi [15]	34
Figure 10:	Product flows in the closed-loop supply chain, adapted from Hamza et al. [16] and Fleischmann et al. [17].....	36
Figure 11:	Firms' strategic reasons for handling returns [7]. The sum of responses exceeds 100% because respondents were able to specify multiple reasons.	37
Figure 12:	Manufacturers' return processing activities, by proportion used, in Hong Kong mobile phone market	38
Figure 13:	Optimal reverse supply chain designs based on proportion of new returns and time-value decay [25].....	41
Figure 14:	Context of our research within Dell's strategic initiatives.....	50
Figure 15:	Reverse supply chain functions within the structural organization of Dell	52
Figure 16:	Reverse supply chain strategies of Dell businesses using Wikner and Tang's structural framework [4]	57
Figure 17:	Collection methods described by Savaskan et al. [49], adapted by Kumar and Putnam [13]	59
Figure 18:	Dell's collection methods	60
Figure 19:	Assessment of Dell's capabilities in the reverse supply chain using Janse et al.'s eight-part framework [3]	62
Figure 20:	High-level process flow diagram at GENCO returns processing facility in Lebanon, TN.....	64
Figure 21:	Pages from the ARM Model User's Guide. We distributed the guide 23 stakeholders—from individual contributors to a vice president—covering ARB, GSP, and GRL in DAO, EMEA, and APJ.....	68
Figure 22:	Dynamics between transformational expenses and recovery in the reverse supply chain, adapted from Jayaraman [35]	70
Figure 23:	Components of a modeling scenario.....	73
Figure 24:	Network flow problem with a time component.....	77
Figure 25:	Flow network with product quality constraints.....	78
Figure 26:	Optimized flow network using downgrade process	79
Figure 27:	Location of capacity constraint limitations	88
Figure 28:	Quality gradient depicting the implicit downgrade process in the ARM model.....	90

Figure 29: Input screen worksheet, consisting of (a) general scenario and product information followed by the “three Rs”: (b) returns, (c) rework, and (d) recovery	91
Figure 30: Production schedule worksheet	95
Figure 31: Depreciation profiles worksheet	96
Figure 32: Recovery channel worksheet	98
Figure 33: Top portion of the nodes worksheet, where we define the 17 nodes in the flow network	101
Figure 34: The network flow worksheet in its entirety, with all ten sections (a) through (j) identified.	102
Figure 35: Formulation worksheet with its five sections: (a) objective function, (b) flow decision variables, (c) rework and channel inventory decision variables, (d) capacity constraints, and (e) flow constraints.....	103
Figure 36: Cost attributions worksheet with its eight sections identified: (a) recovery channel costs, (b) recovery channel rework volumes, (c) recovery channel rework proportions, (d) inventory to channel vs. inventory sources, (e) rework enabled, (f) rework cost per box, (g) rework setup costs, and (h) total costs	105
Figure 37: Recommendations worksheet. The five sections are (a) rework activities, (b) recovery channels, (c) implementation plan, (d) financial summary, and (e) benefit per box.....	107
Figure 38: Dynamic network flow diagram of an example reverse supply chain. Yellow nodes represent inventory types, green nodes represent recovery channels, arrows (arcs) represent enabled transformational activities, and arrow widths convey relative flow volumes.	111
Figure 39: Sensitivity analysis worksheet, consisting of (a) scenarios, (b) PivotChart, (c) data table, (d) PivotTable, and (e) PivotTable sidebar	112
Figure 40: Scenarios section of the sensitivity analysis worksheet	113
Figure 41: For net recovery prediction functions by weekly volume for an undisclosed Dell product family. The functions apply various transformations on weekly sales volume.....	115
Figure 42: Methodology of the small form factor case study	117
Figure 43: Cumulative (4-quarter) network flow diagrams for four alternative SFF reverse supply chain designs: (a) liquidate only, (b) repair capabilities only, (c) complete repair and resale capabilities, and (d) fully enabled with teardown.....	120
Figure 44: Sensitivity analysis plot for key factors in the SFF case study, showing positive net income across all scenarios with 67-119% net recovery	122
Figure 45: Residual plots of SFF sensitivity analysis linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual.....	123
Figure 46: Residual plots of SFF sensitivity analysis second-pass linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual	126
Figure 47: Plot of net income and net recovery moving in opposite directions	127
Figure 48: Sampling of the Alienware product line	128
Figure 49: Methodology of the Alienware repair case study	129
Figure 50: Cumulative (4-quarter) network flow diagram representing the current state reverse supply chain for Alienware repair in EMEA (i.e. no repair option)	131

Figure 51:	Cumulative (4-quarter) network flow diagram with the Alienware repair capability enabled in EMEA	132
Figure 52:	Sensitivity analysis plot for repair cost and price erosion in the Alienware repair case study	134
Figure 53:	Sensitivity analysis plot for refurbished sales capacity and refurbished price in the Alienware repair case study	135
Figure 54:	Residual plots of Alienware repair analysis linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual.....	136
Figure 55:	Residual plots of Alienware repair analysis second-pass linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual	138
Figure 56:	Methodology of the reusable packaging case study	140
Figure 57:	Reusable packaging material network	141
Figure 58:	Screenshot of the reusable packaging worksheet within Dell's CBA spreadsheet. Input fields are denoted with a yellow background.	142
Figure 59:	Reusable packaging options sourced from four different vendors: (a) cardboard box with cardboard interior, currently in use by Dell; (b) solid plastic box with foam interior, (c) corrugated plastic box with foam interior, and (d) air bag placed within paper envelope	144
Figure 60:	Sensitivity analysis of baseline vs. scenario 1 with actual demand and turnaround time factors	146
Figure 61:	Sensitivity analysis of baseline vs. scenario 1 with shrinkage rate and turnaround time factors	147
Figure 62:	Methodology of the teardown dashboard case study	151
Figure 63:	Context of teardown in the flow of parts through the reverse supply chain	152
Figure 64:	Initial mockup for the teardown dashboard, tracking trends and current state for systems and parts by volume, relative volume, deviation from plan, marginal benefit, net recovery, and revenue allocation	153
Figure 65:	Squarified heatmap used to indicate major areas of concern with respect to (a) part failure rates, and (b) demand fulfillment by part. In both cases, rectangles size indicates overall volume and color indicates deviation from optimal: red for excessive failures in (a) and a shortage of parts in (b).	154
Figure 66:	Dashboard architecture.....	156
Figure 67:	Dashboard data control panel in manual refresh mode	157
Figure 68:	Dashboard front-end architecture	158
Figure 69:	Current implementation of the teardown dashboard, showing (a) inventory levels across ARB, GSP, and GRL, (b) part inventory and trends, including top parts scrapped and reclaimed by value and count, and (c) systems torn down by number and value	159
Figure 70:	Demonstration of slicer-based filtering in action on a panel of a dashboard. As a filter is selected, the linked charts and the other filters react. Dark gray indicates a selection, white a deselection; light gray indicates no data available.	161

Figure 71:	Prominence of reverse supply chain concepts in supply chain literature in n-grams. Of the books scanned by Google between 1990 and 2008, proportion of those with the n-gram phrase “reverse supply chain” or “reverse logistics” of the ones with “supply chain” or “logistics” [59].	166
Figure 72:	The functions of Vick’s proposed “EVP of Reverse Logistics” role [60]. The different colors correspond to the fragmented reporting structure of these functions in most firms today.	167

Table of Tables

Table 1:	Reverse supply chain design decisions by process stage	21
Table 2:	Reasons for returns	29
Table 3:	Return conditions	29
Table 4:	Types of activities in each of the “three Rs” in Dell’s reverse supply chain process: receipt, rework, and recovery	30
Table 5:	Waste associated with each of the reasons for product returns	31
Table 6:	Costs associated with a 0% return rate by return reason	32
Table 7:	Common types of mathematical optimization	43
Table 8:	Summary of reverse supply chain optimization models. For comparison purposes, we append the ARM model presented in Chapter 4.	46
Table 9:	Dell’s strategic pillars	49
Table 10:	Reverse supply chain responsibilities across Dell’s groups	51
Table 11:	Regional differences in Dell’s reverse supply chain	53
Table 12:	CLSC building blocks described by Wikner and Tang [4]	57
Table 13:	Summary of Dell’s capabilities in the reverse supply chain	62
Table 14:	Typical focal questions of modeling scenarios	73
Table 15:	Typical product granularities of modeling scenarios, listed from broadest to narrowest	73
Table 16:	Network flow formulation features in the ARM model	75
Table 17:	Model parameters	80
Table 18:	Decision variables	81
Table 19:	Auxiliary variables	81
Table 20:	Overview of the 10 worksheets constituting the ARM model	86
Table 21:	Summary of variables and constraints used in the formulation	88
Table 22:	Explanation of input fields in the modeling scenario section	91
Table 23:	Explanation of input fields in the returns section	92
Table 24:	Explanation of input fields in the rework activity section	93
Table 25:	Explanation of input fields in the recovery channel section	94
Table 26:	Explanation of input fields in the production schedule worksheet	95
Table 27:	Explanation of input fields in the depreciation profiles worksheet	97
Table 28:	Explanation of input fields in the revenue per device section of the recovery channel worksheet	99
Table 29:	Explanation of input fields in the cost per device section of the recovery channel worksheet	100
Table 30:	Descriptions of the sections in the nodes worksheet	102
Table 31:	Descriptions of the sections in the formulation worksheet	104
Table 32:	Explanations of constraint status icons	104
Table 33:	Descriptions of the sections in the cost attributions worksheet	105
Table 34:	Explanation out columns in the rework activities section	107

Table 35:	Explanation of output columns in the recovery channels section.....	108
Table 36:	Implementation recommendations	108
Table 37:	Financial summary fields	109
Table 38:	Explanation of benefit-per-box calculations based on the current recommendation for the rework activity or recovery channel	110
Table 39:	Descriptions of the sections in the sensitivity analysis worksheet	112
Table 40:	Estimated sensitivity analysis run-times	113
Table 41:	ARM model parameters for the smartphone case study.....	118
Table 42:	Impact estimates of four alternative SFF reverse supply chain designs in the U.S.....	120
Table 43:	Input fields and value ranges used to define the modeling scenarios for the SFF returns sensitivity analysis.....	122
Table 44:	Sensitivity analysis of SFF returns based on key factors	124
Table 45:	SFF sensitivity analysis second-pass linear regression results	125
Table 46:	ARM model parameters for the Alienware repair case study	130
Table 47:	Comparison of two alternative supply chain designs for Alienware returns in EMEA: (a) no repair, and (b) repair-with-resale capabilities.....	132
Table 48:	Input fields and value ranges used to define the modeling scenarios for the Alienware repair sensitivity analysis.....	133
Table 49:	Sensitivity analysis of Alienware repair based on key factors	135
Table 50:	Alienware repair analysis second-pass linear regression results	137
Table 51:	Explanation of input fields for the simulation	143
Table 52:	Simulation input parameters for baseline and the three alternative scenarios	145
Table 53:	Three-year NPVs of the four reusable packaging scenarios under different conditions.....	145
Table 54:	Explanation of columns in the Connections table; this table identifies the data sources behind the teardown dashboard	190

Table of Listings

Listing 1:	Network flow line generator macros	180
Listing 2:	Benefit-per-box calculations.....	182
Listing 3:	Sensitivity analysis macros	184
Listing 4:	Implementation of InventoryStore, a class to simulate the flow of shipping material over time.....	186
Listing 5:	Example usage of the simulator class	188
Listing 6:	Dashboard data control panel macros	191
Listing 7:	TableUpdaterController class	192
Listing 8:	TableUpdater class	194

This page intentionally left blank.

Chapter 1. Introduction

The research we present in this paper stems from our six-month engagement with Dell in the latter half of 2010. The centerpiece is an implementation of a mixed-integer linear program (MILP) we call the Asset Recovery Maximizer (ARM). Our original intent with the ARM model was to answer a specific question: what in Dell's reverse supply chain might need to change as it prepares to enter the small form factor markets across the world? Out of this grew a generic, yet robust, reverse supply chain strategic planning and review, along with various extensions to promote continuous improvement in the reverse supply chain.

To protect the confidentiality of Dell's proprietary information, such as the standard cost of SFF devices or its net recovery rates, we do not disclose actual data in this paper. Instead, we use mock data or masked data in our figures, tables, and text that—to the extent possible—preserve the analysis and conclusions we derived from the actual data.

1.1. Overview of Dell

Dell is an American-based multinational corporation in the information technology (IT) industry. With a network of original design manufacturers (ODMs), original equipment manufacturers (OEMs), and third party logistics providers (3PLs), it designs, manufactures, distributes, and sells computer hardware principally through its own online storefront (dell.com) and other retailers such as Wal-Mart and Best Buy. With its 2009 acquisition of Perot Systems, it also provides IT services in areas including software, business process, cloud computing, and support. Since its founding in 1984 by Michael Dell, it has grown considerably; currently it ranks 38th on Forbes' Fortune 500 list with over \$50 billion in annual revenue, third in global PC shipments, and fifth in AMR Research's list of Top 25 Supply Chains. Of its approximately 96,000 employees, 38% of them are U.S.-based. Its major competitors are HP and Acer.

Dell's recent decision to expand its product line with small form factor (SFF) devices is driven by shifting dynamics in the personal computing market. While there is some ambiguity within Dell as to the definition of SFF, we settle on handheld devices with screen sizes of 10 inches (10") or less, which encompasses smartphones and tablets. As pictured in Figure 1, Dell's 2010 SFF offerings include:

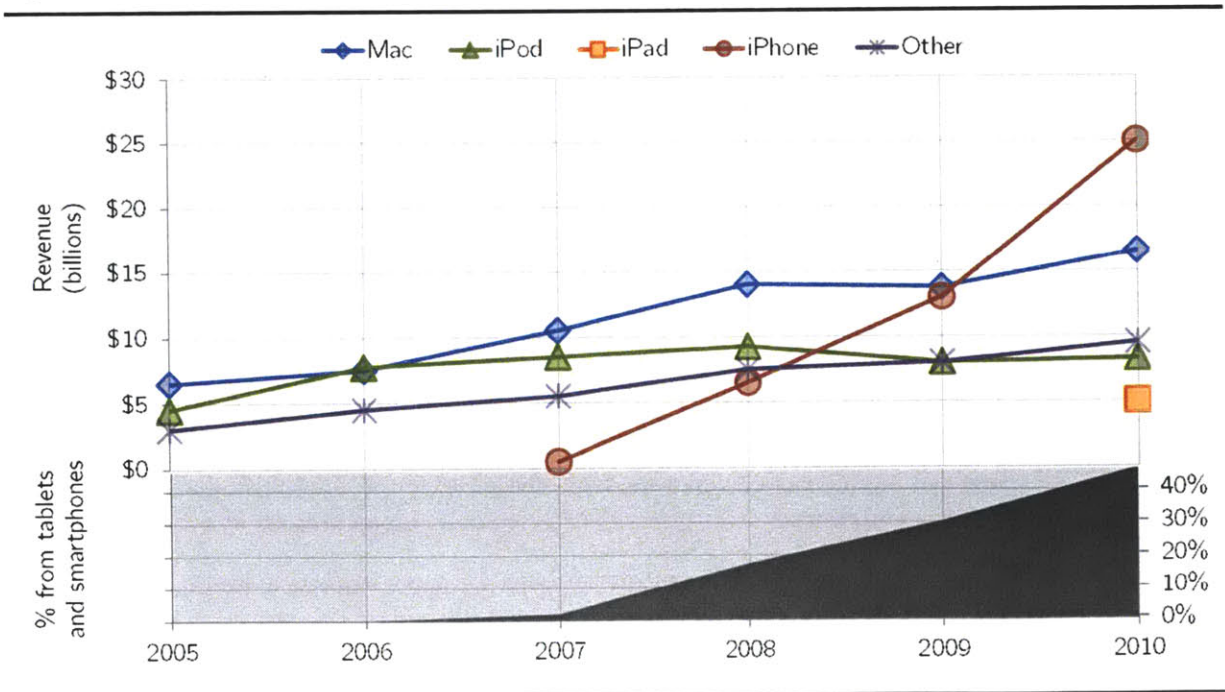
- Mini 3i: 3.5" smartphone with China Mobile and Android operating systems for the Chinese and Latin American markets, respectively,
- Aero: Android-based 3.5" smartphone for the U.S. market,
- Streak: Android-based 5" smartphone/tablet for many markets globally, and
- Venue Pro: Windows Phone-based 5" portrait slider for the U.S. market.

Figure 1: Dell's SFF product offerings to date



Dell has also announced other SFF products in development, with launches planned for 2011. The growing popularity of smartphones and tablets, as well as their potential to displace traditional hardware such as laptops and desktops, is no doubt motivating Dell to make these moves. Consider a Dell competitor who has seen success with SFF for some time: Apple. In Figure 2, we show Apple revenues by product line over the last 6 years based on the company's financial statements. While desktop and laptop sales have proportionally diminished with time, tablet and smartphone sales' contribution to overall revenue at Apple has grown rapidly—from virtually 0% in 2007 to 17% in 2008, 31% in 2009, and 45% in 2010.

Figure 2: Apple revenue over the last 6 years by product line [1]



As Dell extends the capabilities of its supply chain to design, manufacture, and deliver SFF devices, it also is extending its *reverse* supply chain. Its reverse supply chain handles product returns from customers, retailers, and mobile phone service providers. The reasons for returns vary, from customers changing their minds (remorse returns), to defective hardware, to lease expirations, and so on. The treatment of SFF returns differs from that of laptops, desktops, and peripherals—products Dell is expert at handling—for several important reasons. Form factor aside, SFF devices differ in material cost, repair procedures (due to specialized telecommunications-specific hardware), resale channels, and even the business model. In the U.S. smartphone market, for example, most consumers purchase smartphones bundled with a 2-year service contract, and expect phone's purchase price to be greatly reduced—sometimes even *free*—as a result. Refurbished storefronts thus grapple with the challenge of how to price a used device when its new counterpart is essentially given away.

Table 1: Reverse supply chain design decisions by process stage

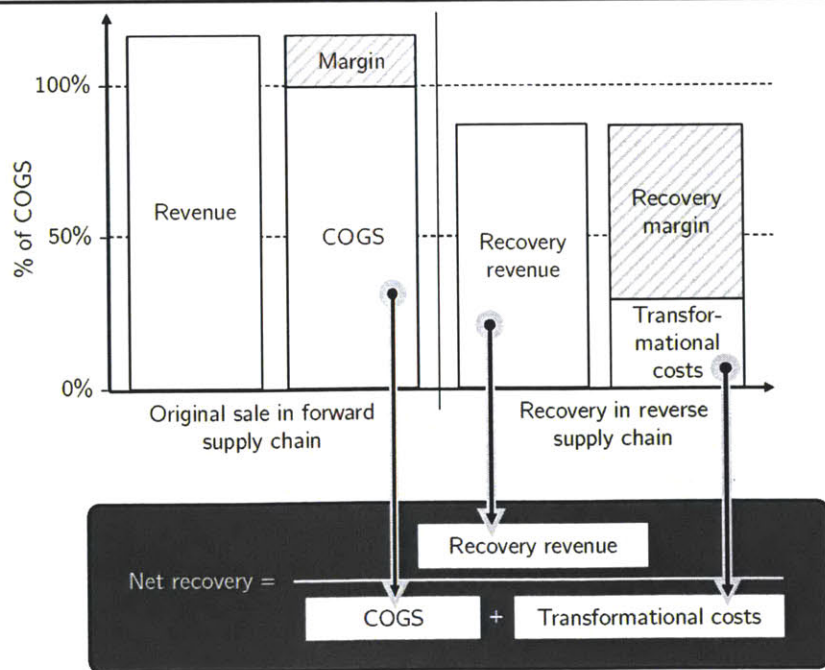
Process stage	Representative questions	Dimensions of answer
1) Receive returns	Where to return?	<ul style="list-style-type: none"> • 3PLs • Location selection
	When to credit customers?	<ul style="list-style-type: none"> • At time shipped vs. time received
	When to send exchanges?	<ul style="list-style-type: none"> • Before shipment or after
2) Rework	Where to refurbish?	<ul style="list-style-type: none"> • Vendor selection • Location selection
	When to dismantle for parts (i.e. "teardown")?	<ul style="list-style-type: none"> • Repair decision rules
	What type of repair quality?	<ul style="list-style-type: none"> • Brand standards • Warranty standards
	What type of transformation?	<ul style="list-style-type: none"> • Locked vs. unlocked • With mobile carrier plan vs. without • With cell card vs. without • With OS vs. without
3) Recover value	Where to resell?	<ul style="list-style-type: none"> • Outlet store • Foreign markets • Developing markets • Refurbishment parts • Service/warranty parts or stock • Scrap
	Where to hold inventory?	<ul style="list-style-type: none"> • Vendor selection • Location selection
	How to hold inventory?	<ul style="list-style-type: none"> • Kitted and boxed vs. unkitted
	How long to hold inventory?	<ul style="list-style-type: none"> • Price erosion • Demand forecast • Holding costs

1.1. Problem Statement

With new product introductions at Dell, we seek an understanding of the key factors that influence reverse supply chain decision decisions, which include those identified in Table 1. The guiding metric when answering these questions is profitability. We hope to find the reverse supply chain design that maximizes recovery revenue at the lowest possible cost to Dell. This problem is not unique to Dell, of course. More generally, distributors and manufacturers—faced with hundreds of billions of dollars of product returns annually [2]—seek an approach to handle product returns most effectively. The optimal solution is not simple. They must design, implement, monitor, and optimize reverse supply chains to handle these flows. The most successful among them minimize returns while simultaneously viewing them as a revenue-generating opportunity, continually seeking out market opportunities to repair, remanufacture, reuse, and recycle returned products. They edge out competitors with these unique advantages.

When discussing profitability, we generally adopt Dell’s preference for the metric of *net recovery*. Roughly, net recovery is the ratio of income to expenses in the reverse supply chain. Income arises from sales revenue as a firm resells its returns sold through the various recovery channels, such as storefronts and auctions. Expenses arise from the material costs of the returns themselves (COGS), the transformational costs involved in repair, teardown, and other reverse supply chain functions, and possibly the related overhead and fixed costs. Figure 3 provides a visual explanation of net recovery using COGS as a baseline.

Figure 3: Visual explanation of the *net recovery* metric. While *recovery revenue* often is less than *COGS* and original *revenue* (as shown), this need not be the case. Bar sizes are for illustrative purposes only, and not reflective of actual values or actual proportions at Dell or elsewhere.



Since *net recovery* is a ratio, we typically speak of net recovery measurements in percentage terms. Break-even net recovery is 100%; this is where the material costs of the product plus any transformational costs is recovered through sales. Anything above 100% equates to additional income for the firm (profit), while anything under 100% translates into an expense (loss). In the computer electronics industry, average net recovery is 28% and the top quintile is 64% [2]. Dell, which has a mature supply chain, lays claim to one of the best net recoveries in its industry.

As a profit-and-loss metric, net recovery is more robust than a simple profitability measurement. With COGS as a baseline, net recovery penalizes operations that do not recover sufficient value from the baseline of standard cost, or do so at excessive transformational costs. It does have drawbacks, however. Consider a device with a standard cost of \$10 with two recovery options in the reverse supply chain: Option 1 requires \$10 of transformational cost and will result in recovery revenue of \$10, whereas Option 2 requires \$20 of transformational cost and will result in recovery revenue of \$20. The net recoveries of these options are 50% and 67%, respectively, suggesting we should prefer option 2. Yet both options yield a net profit of -\$10 to the firm, and option 2 actually requires a higher outlay, so we should really prefer option 1. For this reason, and others, we use profitability instead of net recovery as the metric to optimize reverse supply chain planning decisions in our model in Chapter 4.

Despite its limitations, we still consider net recovery a highly relevant metric for measuring performance in the reverse supply chain. As a normalized metric, it works well for tracking performance over time or comparing performance among regional units. It cannot be the only one, however, since it may lead to poor performance in areas not directly tracked by the metric, such as inventory levels or environmental impact. Supplementary metrics, most of which Dell tracks, include:

- Quality metrics such as burn failure rates and percentage of refurbished systems that fail in the field,
- Cost metrics, such as *cost-per-box*, which can include transformational, inventory, and overhead costs,
- Operational metrics, such as days inventory or refurbishment Takt time, and
- Environmental metrics, such as percentage of parts scrapped.

Theoretically, the net recovery metric reflects most of these other metrics. If scrap rates are high or Takt time is too long, for example, the impact can be felt as lower recovery revenue and consequently lower net recovery. Yet as a single metric, net recovery does not reveal exactly which factors explain the change. Of the available metrics, we consider the cost-per-box metric a suitable supplement. It specifically considers the costs of inventory—including WIP and finished goods—to tell a more thorough story and, in the case of the scenario posed above, guide the organization to prefer option 1 to option 2. Without any supplementary metric, the organization may be led to perform excessive transformational activities relative to the benefits, hold too much inventory, or spend too much on overhead.

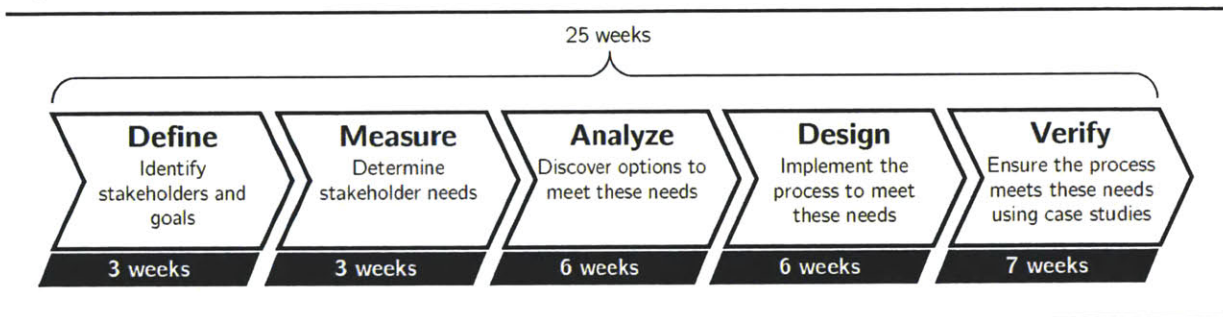
1.2. Hypothesis

We propose that the ARM model informs strategic decision-making regarding reverse supply chain design decisions to maximize net recovery; additionally, it serves as a solid foundation upon which to build management systems for continuous improvement, including cost-benefit analysis calculators and dashboards.

1.3. Research Methodology

In our engagement with Dell, we followed the DMADV (Define, Measure, Analyze, Design, and Verify) six-sigma project execution methodology over the course of 25 weeks, as shown in Figure 4.

Figure 4: Phases and timeline of our six-month engagement with Dell



In the *Define* phase, we worked closely with the head of global reverse logistics at Dell to identify our stakeholders and goals. Our stakeholders were the several groups that hold responsibilities and interests within the reverse supply chain:

- Global reverse logistics (GRL): Physical manages product returns, including shipping and repair, throughout the reverse supply chain.
- Asset Recovery Business (ARB): Handles pricing and sales of refurbished product; also manages the Dell Outlet.
- Global Service Providers (GSP): Handles customer interactions, including issuing return authorizations (CRAs) and warranty repair/replacement.

Our primary goal was to provide a reverse supply chain strategic decision-making model for these three groups. In light of the pending SFF launches, these groups sought a framework for determining the key factors affecting optimal designs in light of uncertainty and continually changing market conditions.

In the *Measure* phase, we formed a working group consisting of representatives from all of these groups across Dell's three regional divisions:

1. North and South America, known as Dell America's Operation (DAO),
2. Europe, the Middle East, and Africa (EMEA), and
3. Asia-Pacific and Japan (APJ).

We collaborated with working group members over conference calls, emails, and visits to some of their sites. We also toured facilities of some of their strategic partners, including the CEVA Logistics merge center and retail fulfillment center in Nashville, TN and the GENCO returns processing facility in Lebanon, TN.

In the *Analyze* phase, we conducted research of reverse logistics closed-loop supply chains (CLSC). This entailed both a literature review and tours of various U.S.-based return processing facilities, including one for Wal-Mart in Waco, TX. For a deep dive in SFF reverse supply chains, we visited an ATC Logistics and Electronics (ATCLE) and TransTrade, both in Fort Worth, TX, which handle repairs for repairs and returns for AT&T, U.S. Cellular, and Blackberry. We also worked with the President of the Reverse Logistics Association (RLA) to issue a Request for Information (RFI) from 3PLs on the repair and recovery of SFF devices. The 40+ detailed responses from providers aided our understanding of these capabilities worldwide.

In the *Design* phase, we implemented the ARM model through an iterative process of development. During the development, we continually interacted with the working group (established in the Measure phase) to ensure all aspects of the model met stakeholders' expectations and needs. Through this consultation, we were able to remedy usability issues, provide detailed recommendations, and expand the power of the tool to encompass a broad range of planning scenarios, from introducing a new product to a single market to augmenting the processing of an entire product line across a broad region.

In the *Verify* phase, we again consulted with the working group to conduct targeted case studies. The case studies demonstrate the value of the model. To ensure Dell continues to leverage its capabilities following these case studies, we assembled a user's guide, led training sessions, and transitioned ownership of the model to two of its internal stakeholders.

1.4 Thesis Outline

In the next chapter, we review prior research in the reverse supply chain space, with a special focus on various types of models used to inform strategic decision-making. We include surveys, diagnostic tools, system dynamics simulations, discrete event simulations, economic models, and optimization models. We summarize the research to identify the areas where the ARM model provides a novel and relevant contribution to the field.

In Chapter 3, we offer an organizational assessment of Dell, and especially the functions that operate within the sphere of the reverse supply chain. The assessment uses diagnostic tools from existing research, including Janse et al.'s capabilities assessment [3], Wikner and Tang's structural assessment [4], and a three lens analysis. We end with our observations of how to drive change in this environment.

In Chapter 4, we provide the ARM model's mathematical formulation. We proceed by describing its implementation, encompassing the input screen, recommendation page, dynamic network flow visualizations, and sensitivity analysis.

In Chapter 5, we conduct our first ARM-based case study, exploring the financial impact of small-form-factor returns in the U.S. over the next financial year. With help from the ARM model, we identify total number of units sold and return rate as the main factor in determining the financial impact of the returns, with price erosion as a secondary factor. Our analysis then quantifies the benefits of various reverse supply chain designs, and we ultimately recommend a design that included multiple recovery channels, including direct-to-customer resale as well as third party auctions.

In Chapter 6, we conduct our second ARM-based case study, first validating the model against the current state of the Alienware reverse supply chain in the EMEA (Europe, Middle East and Africa) region. We then use the model to understand the impact of performing certain types of repairs on returned Alienware systems prior to resale. We identify refurbished price as the main factor in determining the financial impact of the change, with price erosion and repair cost as secondary factors.

In Chapters 7 and 8, we describe the supporting tools that augment the ARM model to promote continuous improvement in the reverse supply chain. Chapter 7 describes a cost-benefit analysis spreadsheet for reusable packaging driven by a discrete event simulator. Chapter 8 details the organization and implementation of an automated, filterable dashboard to provide visibility into system teardown-for-parts decisions.

In Chapter 9, we conclude our research, offering both specific recommendations to Dell and general recommendations applicable to most any firm with a reverse supply chain.

We supplement the chapters with a glossary of terms (Appendix A) and selected computer code listings from our model and supporting tools (Appendices B-D).

Chapter 2. Literature Review

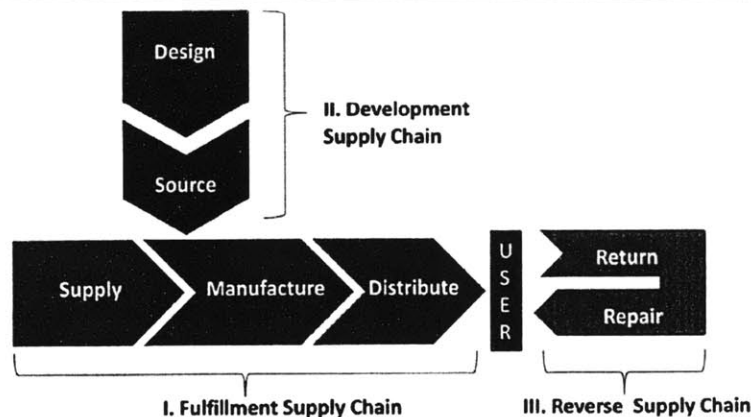
Scanlon identifies *three* supply chains in a firm: the fulfillment supply chain, development supply chain, and reverse supply chain [5]. We show their relationships and processes in Figure 5. The fulfillment supply chain collects raw materials, manufactures products, distributes them, and delivers them to customers. The development supply chain, sometimes viewed as part of the fulfillment supply chain, involves the research, development, and sourcing activities for products. The reverse supply chain is where we focus most of our attention in this paper. In the early 1990s, the Council of Logistics Management defined the reverse supply chain as:

The role of logistics in recycling, waste disposal, and management of hazardous materials; a broader perspective includes all relating to logistics activities carried out in source reduction, recycling, substitution, reuse of materials and disposal" [6].

Rogers and Tibben-Lembke broadens the definition even further, describing it as any aspect of the supply chain where materials are sent "backward," which is not just logistics, but remanufacturing, refurbishing, recycling, obsolete equipment disposition, and asset recovery [7]. We prefer this broad definition. Further, for the purposes of this text we consider *reverse logistics* and the *reverse supply chain* interchangeable terms.

Now consider the term *closed-loop supply chain*. Guide and Van Wassenhove describe closed-loop supply chain management as "the design, control, and operation of a system to maximize value creation over the entire life cycle of a product with dynamic recovery of value from different types and volumes of returns over time [8]". The life cycle of a product naturally involves all three of the supply chains identified by Scanlon. When we speak of the *closed-loop supply chain*, then, we intend to encompass all three supply chains, especially for those instances where we wish to treat the three holistically.

Figure 5: The three types of supply chains [5]



After an overview of closed-loop supply chain trends, we focus on decision-making frameworks that relate to our activities at Dell. We first review industry surveys, then discuss diagnostic tools, system dynamics simulations, discrete event simulations, economic models, and, finally, optimization models. We conclude by identifying the key differences between these frameworks and our research.

2.1. Closed-loop Supply Chain Management Overview

Today the value of product returns exceed \$100 billion annually in the U.S., and \$500 billion worldwide [9], reducing firms' profitability by an estimated 3.8% [2]. Reverse logistics costs may account for up to nine cents for every dollar in sales [2]. The way firms look at these costs vary, and has certainly changed over time. Stock claims most firms look at returns as "costly sideshow" [9], but not all. IBM, for one, helped pioneer integration of product returns into business operations [10]. In 1989, it launched one of the first take-back and recycling programs for end-of-life (EOL) products [11]. This marked a movement up the eco-centric hierarchy of product return handling proposed by Lansink a decade earlier [12], shown in Figure 6. Firms operating higher up the ladder are "greener," for one, but better off economically as well, as they recover more value from each product returned.

Figure 6: Lansink's ladder depicting a eco-centric hierarchy of handling product returns [12]



Most firms have not had much of an environmental incentive to climb Lansink's ladder until recently, as governments has passed laws requiring manufacturers to take back returns. The Waste Electrical and Electronic Equipment Directive (WEEE) directive in the European Union (EU), for example, requires manufacturers of electrical and electronic equipment—including personal computers and SFF devices—to implement such reverse logistics operations free of charge to the customer. Member states started enforcing its regulations as early as 2004. In the U.S., take-back regulations vary by state; under half currently require take-back of computers and mobile phones with regulations enacted as early as 2003. More are pending.

2.1.1 Sources of Product Returns

End-of-life (EOL) products in need of recycling constitute only a small part of product returns, however. We identify and describe the eight principal reasons for returns in Table 2.

Table 2: Reasons for returns

Return reason	Description
End-of-life (EOL)	The customer has finished using the product after it has reached the end of its life. The product may have stopped working beyond the warranty period, or is now obsolete.
End-of-lease	The customer has finished using the product after its lease period has expired. Though no longer boasting the latest technology, the product may have useful life remaining.
Dead-on-arrival (DOA)	The customer receives a product in a non-functional or partially defective state, or the product becomes defective soon after purchase.
Service	The customer initiates a repair or replacement request on a product that has become non-functional or partially defective after the DOA period.
Remorse	The customer regrets a purchasing decision and returns the product soon after purchase, often in unopened or like-new condition.
Retail	The customer uses the product but is still entitled to a refund due to a retailer's policy or governmental regulation. For example, in the U.S. a consumer may return a mobile phone within a 30 day period for any reason.
Upgrade	The customers surrenders the product because he or she wishes to upgrade to newer technology, or, in the case of mobile phones, switch telecommunication carriers using different technology. This is not an end-of-life return because the product is not yet obsolete.
Excess and obsolete (E&O)	Retailers and distributors return inventory that they cannot sell.
Recall	The product has a malfunctioning component, exposing the customer to a potential hazard or poor performance, which needs to be repaired or replaced.

The relative proportions of returns by these reasons vary by industry and region, as does overall return volume. At the extremes, household chemical manufacturers see return rates in the 2-3% range [7] while magazine publishers see return rates around 50%. Computer manufacturers are closer to the lower end of the spectrum, with returns between 10 to 20% of total sales.

The condition of returned products varies as well, regardless of reason. We describe them broadly in Table 3.

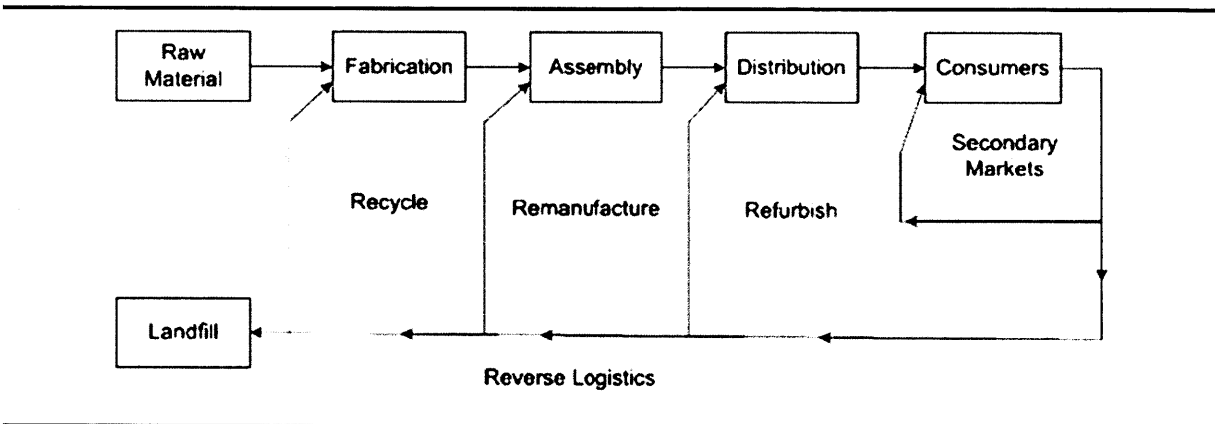
Table 3: Return conditions

Return condition	Description
Unopened/new	The product was returned to the manufacturer without being opened, unwrapped, or turned on.
Opened/new	The product was opened, and perhaps turned on, but may be resold as new. Typically, if there is the potential for customer data on the device, it cannot be treated as new.
Used/like-new	The product has been used, but is in fine working order and without cosmetic defects.
DOA	The product was DOA, and may be eligible for a credit or exchange from the ODM/OEM.
Defective	The product has a cosmetic, physical, or software defect. Manufacturers may rate defects on a scale of severity, such as L1 for superficial scratches and dents to L4 for major component failures.

2.1.2. Processing Returns

Based on the variety of product returns, recycling and/or proper disposal of product returns need not be the only activity of the reverse supply chain. Figure 7 illustrates Kumar and Putnam's description of the flow of returns through the principal activities. Note how a consumer may elect not to return the product at all, opting instead to participate in secondary markets outside the purview of the manufacturer. From the manufacturer's perspective, this may be ideal, since it need not incur any handling costs, though cannibalization of new product sales may be a concern. Alternatively, the consumer may return the product—for any of the reasons mentioned above—in which case the manufacturer guides the returns through a reverse supply chain process flow.

Figure 7: Reverse logistics process described by Kumar and Putnam [13]



Guide and Van Wassenhove describe the process flow in these phases, where product moves from returns collection and management (the front end), to remanufacturing operations (the engine), to remanufactured products market development (the back end) [8]. The reverse logistics team at Dell calls these three phases “the three Rs”: receipt, rework, and recovery. We show the reverse logistics activities corresponding to each of these phases in Table 4.

Table 4: Types of activities in each of the “three Rs” in Dell's reverse supply chain process: receipt, rework, and recovery

Process stage: each of Dell's “three Rs”	Types of activities
1) <u>Receipt</u> : collecting and managing returns	<ul style="list-style-type: none"> • Physical inspection • Hardware tests • Software tests • Customer return authorization (CRA) verification
2) <u>Rework</u> : transforming returns to maximize profit	<ul style="list-style-type: none"> • Repair • Recondition • Remanufacture • Teardown

Process stage: each of Dell's "three Rs"	Types of activities
3) <u>Recovery</u> : selling products, parts, and materials in the most appropriate manner	<ul style="list-style-type: none"> • Use parts for warranty stock • Use parts for repairs • Sell parts • Sell materials as scrap • Recycle materials • Send materials to landfill • Sell refurbished products through outlet • Auction parts, products, or materials • Sell products as new • Send product back to customer • Return to supplier (for credit) • Exchange with supplier

2.1.3. Returns as Waste

The top of Lansink's eco-ladder is "prevention of waste," an ideal state representing a 0% return rate. His preference for absolutely no returns harmonizes with lean manufacturing's philosophy, which targets for elimination anything wasteful, defining waste as those non-value-added activities the customer is not willing to pay for, such as transportation, inventory, motion, waiting, overproduction, over processing, defects, mismatched specifications, and squandered human talent. As Table 5 reveals, each of the reasons for product returns embodies one or more of these wastes. Consider E&O returns, for example. Waste occurs by moving an unsold product to a retailer and back again (transportation waste), storing the product (inventory waste), never selling it (overproduction waste), and perhaps even designing a product people did not want (mismatched specification waste).

While Stock et al. consider minimizing product returns is a worthy goal, they do not consider "zero returns" the optimal policy; at some point, they say, preventing returns is more costly than handling them [9]. We agree, and in Table 6 we offer some reasons why we believe this to be the case.

Table 5: Waste associated with each of the reasons for product returns

Return reason	Waste	Explanation
EOL	1. Transportation	Shipping from the customer back to the manufacturer; typically, customers are not willing to pay for this (some preferring to send the product to a landfill instead), while the E.U. and some U.S. states now require manufacturers to pay for transportation and disposal of EOL returns.
End of lease	1. Transportation	Shipping from the customer to the lease issuer (or designee)
DOA	1. Transportation	Shipping from manufacturer to retailer to customer and back
	2. Defect	Problem triggering the DOA condition
Service	1. Transportation	Shipping from customer to service center (and back to the customer with repair or replacement parts)
	2. Defect	Problem triggering the service event
Remorse	1. Transportation	Shipping from manufacturer to retailer to customer and back

Return reason	Waste	Explanation
	2. Mismatched specifications	Improper advertising, labeling, descriptions, or recommendations, misleading customers into purchasing a product that did not look or function as expected
Retail	1. Transportation	Shipping from manufacturer to retailer and back
	2. Mismatched specifications	Retailer ordering the wrong
Upgrade	1. Transportation	Shipping from customer to manufacturer
E&O	1. Transportation	Shipping from manufacturer to retailer and back
	2. Inventory	Storing the product
	3. Overproduction	Manufacturing a product that was never sold or used
	4. Mismatched specifications	Designing and manufacturing a product with insufficient demand
Recall	1. Transportation	Shipping from manufacturer to retailer to customer and back
	2. Defect	Problem triggering the recall condition

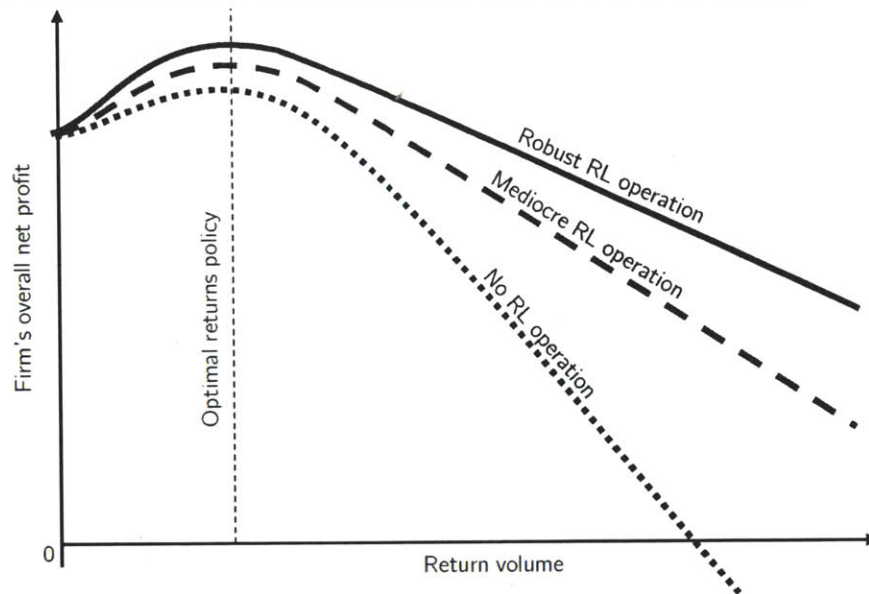
Table 6: Costs associated with a 0% return rate by return reason

Return reason	Costs associated with a 0% return rate
EOI	<p>Products have a finite lifetime. A product designed to last a lifetime seems infeasible for all but the simplest products, like hammers. Technology products quickly become obsolete, and while it might be possible to extend the lifetime of products through upgradable hardware or software, extensibility has a cost and can only take aging technology so far.</p> <p>Further, once a product has reached its end of life, government regulations are now compelling firms operating in certain regions to take these products back. Non-complying firms face stiff fines.</p>
End of lease	<p>The appeal of leased equipment as an alternative to outright purchases would be severely undermined if leases could not be returned. There are several possible ramifications: 1) leasing contract lengths would always have to match the useful life of the equipment, 2) customers would be forced to purchase leased equipment at its residual value at the end of the leasing term, or 3) leasers would abandon equipment despite any residual value. Regardless of the scenario, the net result is the same: leases become more expensive for customers, thus they demand fewer leases, and the leaser leases less. The bottom line impact to profit could be significant.</p>
DOA	<p>The cost of ensuring zero shipped defects—far beyond the standard of six sigma—would be very high. A manufacturer would need the finest materials, the most robust in-house quality assurance testing, and a firm grip on all forward logistics from the plant to the retailer and/or customers hands to ensure no damage along the way.</p>
Service	<p>The cost of ensuring zero defects while a product is under warranty, through the gamut of possible foreseen and unforeseen conditions, would also be very high, if even feasible.</p> <p>Narrowly scoped limited warranties, or the absence of any sort of warranty would likely be prohibited by law, hinder the value of the product in the customer's eyes, or both.</p>
Remorse	<p>Despite proper labeling, expectation-settings, etc., some customers nonetheless later reconsider their purchasing decisions due to a changing financial position, personal needs, or other circumstances beyond the firm's control.</p> <p>For some sales channels, like online stores or over-the-phone catalog orders, the customer, unable to touch and try the product, may be reluctant to make a purchase without having the option to return. Having a strong no-returns policy would inhibit sales [14].</p> <p>Again, by law, firms are compelled to accept customer returns within a particular window for any reasons, such as the 30-day return rule for mobile phones. Non-complying firms face stiff fines.</p>

Return reason	Costs associated with a 0% return rate
Retail	<p>Manufacturers who sell to retailers would certainly prefer a zero-returns policy, but they may not have the market power to strong-arm retailers into accepting such a condition, especially if that retailer is Wal-Mart. By not distributing products through retailers who accept a zero-returns policy, a manufacture may miss out on tremendous sales volume.</p> <p>Further, retailers may prefer to order extra inventory and send the rest back as E&O. With their salvage costs for E&O likely higher than a manufacturer's, which would presumably be better positioned to handle E&O for the specific product they manufacture better than the retailer, the newsvendor-based solution, the retailer would be expected to order less and the likelihood of a stock out would be higher. This would yield in a lower expected profit for the manufacturer.</p>
Upgrade	<p>For shorter-lifecycle products where customers have an expectation of an upgrade option, manufacturers are essentially compelled to provide one by analogy with the Nash equilibrium of Bertrand competition. Otherwise, its existing customers, who may not wish to replace a growingly obsolete product, may switch to a competitor. In some real world examples, such as T-Mobile, a company even extends the upgrade opportunity to competitors' products to further motivate this switchover.</p>
E&O	<p>Similar to the explanation for retail returns, as a newsvendor inventory-setting framework would suggest, some E&O is desirable and expected when there is salvage value – and the salvage value increases directionally with the sophistication of a manufacturer's reverse logistics capabilities. If a firm was careful to manufacture products so as to guarantee no E&O inventory, it likely would producing less than the profit-maximizing quantity.</p>
Recall	<p>The cost of ensuring zero recallable defects—far beyond the standard of six sigma—would be very high, especially for defects that stem from unforeseeable operating conditions or extended use.</p>

Figure 8 summarizes the point that while returns are a form of waste, zero returns is not the optimal return rate. Considering the costs associated with achieving no returns, firms are better off reducing returns to a sweet spot—the optimal return policy—where any margin effort in reducing the return rate would not be cost effective. This cements the importance of the indefinite existence of a reverse supply chain, and, further, that such a group has the ability to act as a *profit center*. In Figure 8, we show three lines for varying degrees of reverse logistics capabilities at a firm: none, mediocre, and robust. These lines correspond, roughly, to degrees of sophistication in the reverse supply chain. In the *No RL operation* line, the base case, there is no serious reverse logistics function to speak of; here, everything is simply recycling or sent to a landfill in accordance with regulatory compliance. The two other lines show increasing sophistication as the firm works its way up Lansink's ladder, where the returns are perhaps auctioned to third-parties for refurbishment and resale, or, alternatively, refurbished or remanufactured in-house where possible and resold at a higher value. These activities cost the firm money, certainly, but the net outcome is additional profit.

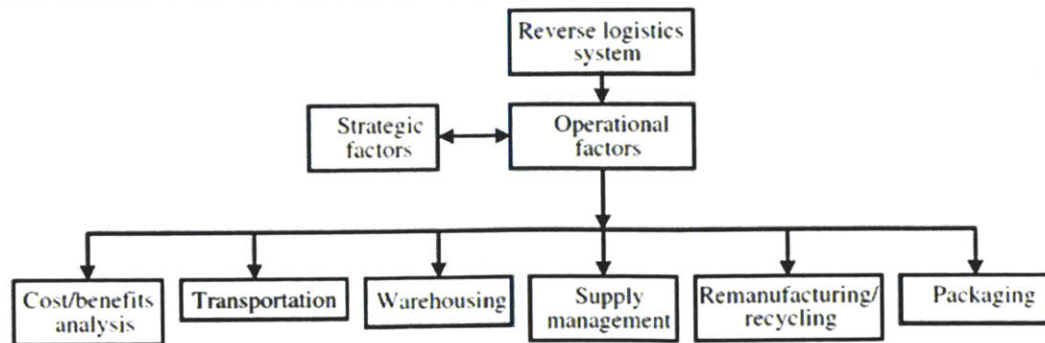
Figure 8: Effect of reverse logistics on profitability across various return policies



2.1.4. Reverse Logistics Functions

Given the variety of reasons for returns, and conditions in which return products arrive, Dowlatshahi identifies the key logistics functions needed to handle them properly, shown in Figure 9 [15]. He cites many operational factors, including cost/benefit analysis (CBA), transportation, warehousing, supply management, remanufacturing/recycling, and packaging. We incorporate all these factors in the ARM model discussed in Chapter 4. Some firms, including as Dell, outsource one or more of these functions to 3PL providers. Dowlatshahi cautions that these factors alone do not describe the entirety of the reverse logistics ecosystem. He notes interdependencies with other areas of the firm, too, including finance/accounting, customer service, quality/reliability, purchasing, and design/engineering.

Figure 9: Aspects of reverse logistics from Dowlatshahi [15]

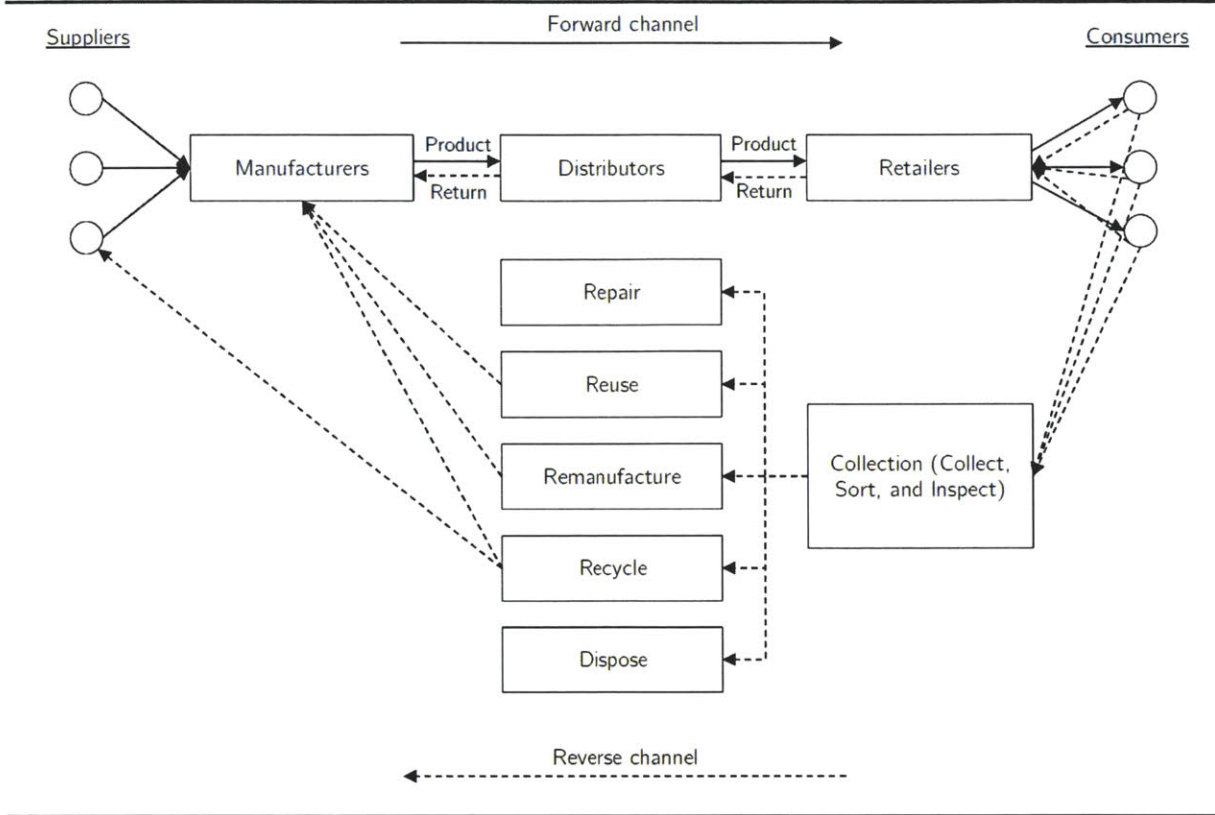


In Figure 10, we adapt the work of from Hamza et al. [16] and Fleischmann et al. [17] to show the reverse logistics process in the context of the closed-loop supply chain, showing both forward and backward product flows simultaneously. In the far right of the diagram, in the forward flow, we show a “few-to-many” distribution of products: products flow from a small number of distributors to a larger number of retailers to a still larger number of customers. At the same time, we show a “many-to-few” recovery of products: returns flow from a large number of consumers to a few collection centers.

To achieve optimal efficiency, Fleischmann et al. suggest optimizing considering the few-to-many and many-to-few problems of the respective forward and reverse flows simultaneously. This turns out to be more challenging than it may seem. For example, the authors note difficulties in applying traditional inventory models to the reverse supply chain:

1. The timing and volume of returns are generally controlled by consumers and retailers, meaning, unlike in the forward chain where the firm can purchase a specific quantity of materials, here the firm must accept the hand it is dealt.
2. The reverse supply chain is squeezed between the “push” of returns and the “pull” of demand for refurbished products or replacement parts, leaving it with a two-echelon inventory management challenge.
3. Inventory is not monotonically depleted as parts or products are processed, since customer returns continually contribute to new inventory simultaneously, raising mathematical challenges for the mathematical models underlying inventory management tools.
4. The necessity of tests to determine the condition of parts expose the remanufacturing process to increased complexity not seen on the forward side, as does choosing the most economical tasks among feasible options. This makes usage of traditional MRP (material requirements planning) systems inadequate.
5. Unlike the forward flow, there are decision-points to determine how the inventory is used. It may be repaired, scrapped, harvested for parts, etc. depending not only on external factors such as demand but the condition of the specific product or part in inventory.

Figure 10: Product flows in the closed-loop supply chain, adapted from Hamza et al. [16] and Fleischmann et al. [17]



2.1.5. Strategies for Effective Management

Managing the reverse supply chain is a complex undertaking, for sure, but there are rewards for doing it well. Stock suggest “the process – which may include the remanufacturing, refurbishing, recycling, reuse, or disposal of goods – should be seen as an opportunity to build competitive advantage” [9]. He claims firms that give reverse logistics operations its own turf—meaning separate space from forward operations, which may include a 3PL relationship—have a greater ability to build out the reverse supply chain efficiently and fully exploit recovery channels, including newer online marketplaces such as eBay.com and FastAsset.com. In general, reverse logistics operations yield lower-price-point products that can create new market opportunities.

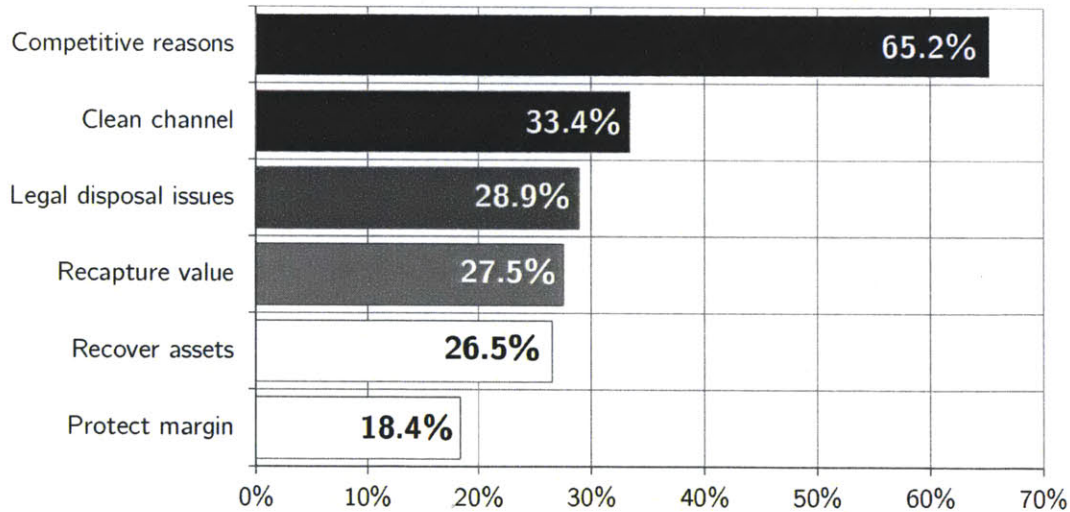
Wikner and Tag identify the major obstacle to closing the loop on the supply chain is the difficulty in building an efficient production control system that handles uncertainties in the returns process [4]. They do not see researchers—or software developers—stepping in to solve the unique problems related to reverse flows. They propose a customer order decoupling point (CODP) approach, which researchers have used successfully in other areas. Given the four building blocks of forecast-based and demand-based manufacturing and remanufacturing products, the authors arrive at 15 different closed-loop supply chain designs that can serve as the framework to strategic supply chain decision-making. We use these building blocks to describe Dell’s supply chain in Section 3.3.1.

The Aberdeen Group proposes additional recommendations for an effective reverse supply chain: ensure the group has oversight and accountability, upgrade its IT systems, recover more costs from suppliers, and close the loop among service/warranty, sales, marketing, design, and manufacturing organizations to eliminate the type of waste that lead to higher return rates [2].

2.2. Surveys & Trends

In 1999, when reverse logistics was a burgeoning new field, Rogers and Tibben-Lembke used survey responses from 311 reverse logistics managers (~30% response rate) to conclude that many firms had started to realize the importance of reverse logistics as an important part of their business' mission, and, for some, a strategic tool for competitive advantage [7]. Respondents identified other, more pressing issues; company policies; lack of IT support; and competitive issues as the key barriers to developing a better reverse logistics program. We show the strategic reasons they cited for processing returns in Figure 11. Competitiveness, the top reason, outranks the next highest one by a two-to-one margin.

Figure 11: Firms' strategic reasons for handling returns [7]. The sum of responses exceeds 100% because respondents were able to specify multiple reasons.

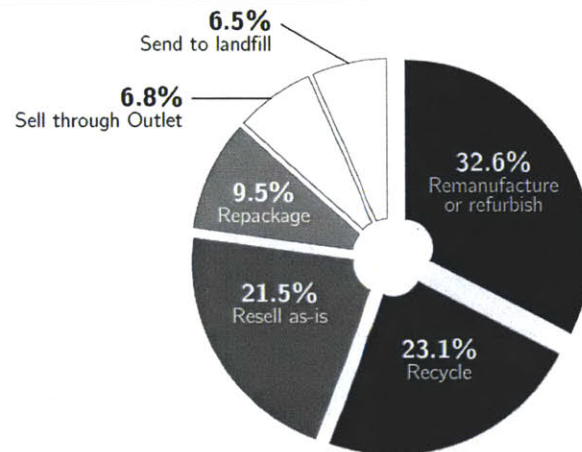


Rogers and Tibben-Lembke also find the average estimate of product returns' effect on net profit ranged from 3.0% – 4.4% (at a 95% confidence interval). Further, while a reverse logistics operation isn't the primary basis upon which a firm operates, doing so efficiently has been shown to offer a competitive edge of up to 0.6% of revenue [18].

Chan and Chan followed up on Rogers and Tibben-Lembke's research with a survey of their own in the Hong Kong mobile phone market [19]. Despite the seemingly greater relevance of reverse logistics in a for a short lifecycle mobile phone product with high disposal rates, they discovered the same barriers

to reverse logistics development. The low priority of reverse logistics relative to the other issues is still the major barrier in realizing reverse logistics systems. In this industry, firms are motivated to recapture value (more than 20% of returns could be resold as-is), use reverse logistics as a competitive advantage (as the field is in its infancy and some competitors do not have solid reverse logistics operations yet), and be a good corporate citizenship (to gain publicity). Key activities undertaken by surveyed firms are shown in Figure 12.

Figure 12: Manufacturers' return processing activities, by proportion used, in Hong Kong mobile phone market



Genchev performed a case study at Tech Data, a Fortune 500 company acting primarily as a wholesaler of IT products [20]. Rather uniquely, they maintain their reverse logistics operations in-house instead of outsourcing to a 3PL as Dell does. Upon receipt of returns, there is an intensive inspection process undertaken by highly trained engineers—to ensure optimal routing. Tech Data focuses on time-to-customer-credit as a key metric to track. The key recommendations from the successful implementation are as follows:

1. Gain senior management support and turn reverse logistics into a company-wide initiative;
2. Involve your customers in the reverse logistics design process;
3. Give distinct recognition to the employees involved in handling returns within the firm;
4. Implement carefully developed written rules and procedures that reflect both internal and external concerns; and
5. Assign strict responsibility for the execution of the reverse logistics program.

Daugherty et al. offer another recommendation: invest in your firm's information technology capabilities, as you may greatly enhance the economic performance and service quality of its reverse supply chain by doing so [14]. Investments may include MRP systems, ERP systems, RFID, and ARP. Based on 118 surveys from members of the Automobile Aftermarket Industry Association (AAIA), many re-

spondents indicated their reverse logistics operations were underfunded, and a significant correlation appeared between investment in IT capabilities and both economic performance and service quality.

Scanlon surveys 75 senior managers of original equipment manufacturers (OEMs) in the mobile phone industry, including Nokia, Samsung, and Motorola [5]. Though statistically significant findings were problematic since only 21% of participants finished the survey, the author finds that high performers considered reverse logistics while designing the forward supply chain. OEMs competing on innovation rather than price tend toward supply chain designs that are more flexible than efficient. Correspondingly, those firms have a more regional-based remanufacturing strategy and outsourced less of their activities to 3PLs.

2.3 Diagnostic Tools

Janse et al. use surveys, academic experts, and PwC consultants to arrive at a diagnostic tool for reverse logistics systems [3]. The goal of the tool is to reveal true costs, revenues, and end-to-end performance in a manner while promoting cross-departmental collaboration within the company. The framework measures various dimensions, including alignment with business objectives, from a scale of one (immature) to four (mature). Broadly, they consider the mature companies the ones who treat reverse supply chain operations as an integrated profit center rather than an isolated cost center. The diagnostic tool helps companies move towards a more mature approach, especially helpful if they are operating in the large gray area between these two extremes.

Based on industry surveys and research into strategic decision-making factors, Subramoniam et al. define an approach to making reverse supply chain decisions: the remanufacturing decision-making framework (RDMF) [21]. When making decisions, they propose considering the following aspects: economic, environmental and social impact; design for product lifecycle costs; IP; recovery value; customer specifications; disposal costs; brand erosion; green initiative; government regulations. The framework helps firms prioritize their initiatives, such as ensuring products are designed for remanufacturability, ensuring the accuracy of environmental labeling, avoiding certain hard-to-dispose-of substances in products, and designing with minimal material and energy usage.

2.4 System Dynamics Simulation Models

Spengler and Schröter are two of the initial researchers to apply system dynamics principles to the reverse supply chain, in 2003 [22]. Their goal is to help a manufacturer of imaging electronics determine an optimal spare parts inventory management strategy by incorporating parts from product returns. They run multiple simulations across various production volumes to determine the quantity that maximized gross profit. While the conclusions are narrowly scoped, they identify feedback loops and delays upon which other researchers conduct further research.

Tan and Kumar build a system dynamics model for a computer manufacturer to evaluate reverse supply chain alternatives [23]. Their model's 92 input variables include the quality, volume, and type of returns; variable costs such as transportation, repair, scrapping, and storage; and, importantly, time delays among the steps in the reverse supply chain. The model's key output is an estimation of gross profit. By running multiple simulations with various reverse logistics designs, they determine optimal recovery activities, recovery facility locations, modes of transportation, and resale pricing. With respect to their specific case study, Tan and Kumar associate repair, transportation, sorting, sales, and supplier delays with lower profitability. They also determine return volume had negligible impact on margin, despite minimum flat charges for transportation.

Georgiadis et al. assemble a more comprehensive causal loop diagram by incorporating inventory and capacity in remanufacturing, service, inspection, collection, and distribution areas [24]. The structure of their model assumes the return volume will impact these inventories and capacities, but only gradually and with some delay. In order to use this model, one must exert some effort in estimating the corresponding delay parameters. By using Taguchi's parameter design methodology and ANOVA analysis, the authors measure the net present value (NPV) of gross profit, and determined optimal designs are robust to total demand, which is good since it can be hard to estimate, especially for products with short lifecycles. For products with shorter lifecycles, such as mobile phones, they suggest building and shrinking capacity more aggressively based on recent return rates.

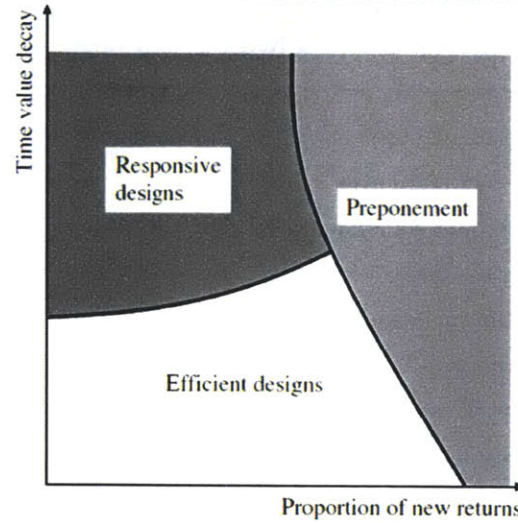
2.5 Discrete Event Simulation Models

Fleischmann et al. develop a simulation of serviceable parts fulfillment sourced in three ways: new orders, repair orders, and dismantling of returns [10]. They run a simulation with seven scenarios based on whether dismantling was enabled, how parts were stocked (push vs. pull), and how part fulfillment decisions were made (optimally, reactively, or forecast-based). The reactive policy's costs grow much faster than the other policies as demand for parts increased, though the forecast-based policy remained close to optimal. They recommend using accurate forecasts to determine the proportion of components to dismantle. Strikingly, they find benefit to dismantling as much as 90% of this flow, after which holding costs and scrap charges began to outweigh the benefits of purchasing new parts.

Guide et al. examine HP inkjet printers and Bosch power tools, with product lifecycles of 18 months and 6 years, respectively [25]. With multiple runs of a simulation model incorporating return rates, sales rates, wait times, among other factors, they determine two principal factors affecting the efficiency of the reverse logistics network: time-value decay and proportion of new returns. While a centralized design is appropriate when time-value decay and proportion of new returns are both low, perhaps counterintuitively, a decentralized "preponement" design, with distributed return/rework facilities, is better when time-value decay and proportion of new returns are both high [26]. When just time-value decay

is high, a responsive design, that is, one that uses more expensive shipping options or collocates facilities, is appropriate. We provide Guide's plot of optimal designs in Figure 13.

Figure 13: Optimal reverse supply chain designs based on proportion of new returns and time-value decay [25]



Hamza et al. propose a general framework of interval-based simulation to estimate TCO in a closed loop supply chain, with probability bound analysis (PBA) replacing real parameters with interval parameters to quantify uncertainty [16]. Separately, Hanafi et al. use demographic data and historical sales of a relevant product in a certain location to predict return rates by location and determine the best locations to collect end-of-life products [27]. They base their simulation on a fuzzy colored petri net, which depicts the flow of product throughout a graph of the reverse supply chain, and explore its utility using the mobile phone market in Australia as a case study.

2.6. Economic Models

In the realm of economic modeling, Kaga finds that a technique called real options analysis can be a good strategic tool, useful for making intelligent reverse logistics design decisions [28]. For his case study, he focuses on a circuit board manufacturer that must decide whether to continue repairing defective boards for customers, or rather replace them with new boards. Due to declining yet volatile prices for new boards, the decision is not clear. With real options analysis, Kaga is able to not only quantify the present value of various design choices, but determine how much a company would be willing to pay to switch the design later, i.e. the option price. Kaga believes as companies' reverse supply chains grow increasingly distributed and complex, the additional uncertainty will result in increased use of real options analysis.

Lin creates a tactical model for Dell to make dismantling decisions for their computer returns [29]. Typically, Dell had opted to repair computer returns, but Lin explores the possibility of instead dismantling.

ting some of those returns and using the parts for various purposes: repairing other systems, servicing customers' warranty needs, and selling the parts through various channels. With part demand growing in these areas, value of the parts individually can be worth more than their value collectively as a self-contained computer—but only to a point. Once part inventory is saturated, the remaining parts are scrapped as excess. Lin's Systems Optimization Routing Tool (SORT) examines the marginal benefit of successive dismantling operations—called *teardowns*—to determine the appropriate quantity to tear down on a monthly basis. One important observation from the case studies is how precipitously the marginal recovery of teardowns drops when teardowns exceed the appropriate quantity. He finds it especially important, then, to accurately account for system and part demand when making teardown decisions, and err on the side of tearing down too little.

Geyer and Blass provide a technique for dealing with uncertainty in reverse logistics processes [30]. Using the recovery of spacecraft circuit boards as a case study, they construct a Bayesian belief network (BBN) to evaluate various reverse supply chain network designs. These designs incorporate varying levels of tests—at environmental, component, electrical, and functional levels—along with repair and remanufacture activities. Given some known probability intervals, such as the likelihood of the returned product functioning correctly is 97-99% given that the environmental test passed, and applying Bayes' law, they establish a probabilistic range of the overall success rate for the network. They use this information to determine these two things: first, whether the design meets requirements by ensuring the lower bound is higher than the minimum tolerable threshold, and second, the reliability of the network, where a smaller interval would indicate greater reliability.

2.7 Optimization Models

Akçalı et al. review 11 reverse supply chain optimization models and 11 closed-loop supply chain models, describing the primary goals of all models as facility location and routing decision-making, sometimes mixed with additional relevant design and/or operational issues [31]. In this section we review the 12 most relevant optimization models to our research, which include a few of the 22 covered by Akçalı. To review, these models are expressed problems in terms of a mathematical formulation, where computer programs are then used to find optimal—or near-optimal—solutions based on real-world data. Each formulation is expressed in terms of an objective function (what needs to be optimized), decision variables (the solution being sought), and constraints. Table 7 summarizes the types of models we review.

Table 7: Common types of mathematical optimization

Optimization method	Abbreviation	Description
Linear program	LP	Objective function and constraints are all linear; decision variables are allowed to assume non-integer values
Integer linear program	ILP	LP, but all decision variables are constrained to be integers
Mixed integer linear program	MILP	LP, but some decision variables are constrained be integers
Mixed integer non-linear program	MINLP	Objective functions and constraints need not be linear. Some decision variables must be integers. This type of formulation may have local optima, making it difficult to determine whether the identified solution is the global optimum.
Fuzzy goal programming	FGP	A linear or non-linear program where the objective function is one of the decision variables, used to find a balance among multiple objectives.

Fleischmann et al. formulate a generic model to determine the optimal design of a reverse logistics network involving disassembly centers, reprocessing factories, and distribution warehouses [32]. Using copier remanufacturing and paper recycling case studies, they find that, in most cases, a product recovery network can leverage a preexisting forward logistics network without deviating from the optimal solution by more than 20%. The authors attribute this to the high correlation between demand and return volumes. This is not the case, however, when transportation costs pull the forward network closer to supplier or raw materials and the reverse network closer to customer.

Willems et al. devise a simpler model to determine, given a fixed set of facilities, the degree to which disassembly of returns for part reuse should be undertaken [12]. The model considered returns of two different qualities: functioning or broken, where broken returns would have to be repaired to be reused. They find that for products with relatively shorter lives and lower initial values (e.g. a water kettle), disassembly did not make economic sense. By varying the cost of repair and disassembly, which they used as proxies for disassembly time, they determine that for these types of products a time reduction of up to 95% does not change the outcome. For medium- and large-sized products, however, disassembly became practice with time reductions of 75% for medium-value products and 65% for high-value products.

Salema et al. build on [32] to analyze multiple products in the forward and reverse logistics networks simultaneously [33]. Critically, they separate factory-to-warehouse-to-customer decision variables into two sets of decision variables: factory-to-warehouse and warehouse-to-customer. They determine optimal solutions can be discovered to this two-tiered formulation much faster than the original one. In essence, this work serves as an example of a general-purpose network flow formulation.

Guide et al. craft a multi-period network flow model to evaluate the time-sensitive nature of handling computer returns at Hewlett-Packard [34]. In each month, they define a subnetwork to track inventory by age, and considered two alternatives: in-house low-touch refurbishment and ODM refurbishment. By varying ODM lead time and the proportion of devices subject to in-house refurbishment, and assuming price erosion of \$20/month, they are able to quantify the benefit of shorter cycle times. They estimate a

boost in profitability by 30 – 50% simply by testing returns up-front to determine the best disposition channel. While the model is not broadly applicable, it provides an approach to creating generic multi-period network flow linear programming formulation.

Jayaran describes his work on RAPP—Remanufacturing Aggregate Production Planning—a decision support module that determines how to allocate labor hours towards the activities of receiving, disassembling, disposing, and remanufacturing mobile phone returns of varying quality levels [35]. He considers six different quality levels, ranging from one (passes all tests) to six (failed a test or did not power up and has some damage). Oddly, his formulation is geared towards minimizing cost rather than maximizing profitability; he does not acknowledge that a higher-cost solution may yield better profitability, suggesting that the organization in which he is deploying the model considers returns more of a cost center than a profit center; this organization would likely score relatively low with Bastiaan et al.’s reverse logistic diagnostic tool [3].

Lieken and Vandaele extend the facility-selecting MILP formulations—such as Willems [12] and Fleischman [32]—by adding a queuing component to account for lead time, inventory positions, and uncertainty [36]. They do this rather elegantly by incorporating the queuing aspects into the objective function only, in the form of inventory holding costs. They arrive at these costs by assuming a G/G/1 queuing model is appropriate for each facility, then approximate the corresponding waiting time, and use Little’s Law to determine yearly inventory costs. Inexplicably, the revenue side of their objective function ignores the fact that a portion of inventory is in a queue and may not be saleable. Despite the imperfections, the model provides some insights. For one, it shows the results are generally equivalent to their non-queuing counterparts when inventory costs are low relative to other costs, such as rework. In those cases where inventory costs are relevant, however, finding optimal solutions can be tricky as the non-linear solution space may be extensive with local optima, as with most non-linear formulations. The authors propose a technique—differential evolution—for finding good approximations within an acceptable time limit.

Min et al. provides a MINLP formulation for determining which collection points to select, with features to determine the facilities to which customers should send their returns [37]. They determine that making appropriate location and allocation decisions for collection points is a key to the success of reverse logistics operations. A year later, Min and Ko later extended this work by adding decision variables to determine if and when to expand warehouses, and to work with multiple products [38]. As NP-hard MINLP formulations, both models rely on a smart algorithm to find approximate solutions within an acceptable time; in both cases, the authors chose a genetic algorithm.

Tsaia and Hung propose a fuzzy goal programming (FGP) formulation with multiple objective structures touted to promote a green supply chain (GSC) [39]. In truth, the model is well suited for any mix of financial and non-financial measures, including activity-based costing (ABC), green production and disposal, long-term strategic activities such as total cost management (TCM), total quality management

(TQM), total risk management (TRM), and total environmental management (TEM), and non-value-added (NVA) activities. Instead of maximizing profitability or minimizing cost, the model attempts to minimize the deviation from the targeted desirability among these measures, weighted by a company's own priorities.

Pishvae et al. enhance the models of Jayaraman, Min [37], and others by proposing simulated annealing as a technique to find near-optimal solutions of a multistage reverse logistics network [40]. Since such network design problems are NP-hard, the computational time increases exponentially as the number of candidate nodes in the network grows, making it impractical to find optimal solutions. Through simulated annealing, an algorithm inspired by the physical process of cooling, they are able to find solutions within 10% of optimal in less than 10% of the time. For larger problems requiring 12 hours of computation, this is a marked improvement, especially in cases where near-optimal solutions are acceptable.

Salema et al. propose a comprehensive model designed to provide both strategic and tactical recommendations [41]. At the strategic level, the model chooses facilities in a four-echelon structure which includes factories, warehouses, distribution centers, and sorting centers. At the tactical level, it estimates the flow of product among chosen routes. To accomplish this, the model's formulation employs two interconnected time scales for this simultaneous strategic and tactical planning. Users of the model can customize the time scales to suit their needs, opting for years at the macro level and months at the micro level, for example, or perhaps a month/day combination. Naturally, their formulation links both time scales together for consistency. The authors acknowledge that for larger problems, such as 25 candidate nodes, five macro periods, and four micro periods, the problems become too complex to solve optimally within acceptable time and memory constraints. They leave more efficient constructions, such as Benders decomposition, for future research.

Lee and Dong provide an approach to solving these seemingly intractable large-scale logistics flows encompassing both forward and reverse flows: a two-stage heuristic algorithm [42]. In the first stage, depot locations are chosen randomly and validated by running the simplex algorithm to ensure product flows without violating constraints. In the second stage, a tabu search is used to improve the routing of product, which works much like simulated annealing, that is, by slightly altering an existing solution slightly to find a nearby better one. The two-stage process repeats within a time bound, and the best solution is reported. In the larger test problems, the approach provided solutions within 12% of optimality in under 15% of the running time of the simplex algorithm.

Table 8 summarizes our review of optimization models for reverse supply chain planning. Despite the large number of models available, we agree with Akçali et al. in seeing a need for more generalized models with demonstrated practical impact and adaptability to uncertainty and risk [31].

Table 8: Summary of reverse supply chain optimization models. For comparison purposes, we append the ARM model presented in Chapter 4.

Author	Decision variables	Objective function	Product variety	Multi-period support	Classification
Fleischman et al. [32]	<ul style="list-style-type: none"> Product flows to various plants and disposition channels: recovery or disposal Which plants, warehouses, and disassembly centers to open (binary) Percentage of unsatisfied demand Percentage of returns uncollected 	Minimize cost	Single product	No	MILP
Willems et al. [12]	<ul style="list-style-type: none"> Product flows to various plants and disposition channels: reuse, recycling, or disposal 	Maximize profit	Single product with 2 quality levels	No	LP
Salema et al. [33]	<ul style="list-style-type: none"> Product flows to and from various factories, warehouses, and customers (closed loop) Amount of unsatisfied demand Which facilities to use (binary) 	Minimize cost	Single product (with extension for multiple products)	No	MILP
Guide et al. [34]	<ul style="list-style-type: none"> Product flows by age class to testing, EMR, or various disposition channels: ODM, broker, secondary market 	Maximize profit	Three products	Yes, for ODM handling cycle time and return volumes	LP
Jayaraman [35]	<ul style="list-style-type: none"> Product flows by quality level that are received, disassembled, disposed, remanufactured 	Minimize cost	Single product with 6 quality levels	Yes, for return volumes only	LP
Lieckens and Vandaele [36]	<ul style="list-style-type: none"> Product flows to various facilities and reuse markets Which facilities to use (binary) Percentage of unsatisfied demand Percentage of returns uncollected 	Maximize profit	Single product	Yes, for queuing only	MINLP
Min et al. [37]	<ul style="list-style-type: none"> Product flows to centralized return center Maximum holding time at each collection point Which collection points to use (binary) Which customers are assigned to each collection point (binary) 	Minimize cost	Single product	Yes, for return volumes only	MINLP with genetic algorithm
Min and Ko [38]	<ul style="list-style-type: none"> Product flows to customers from a warehouse or repair facility Product flows from customers to repair facilities Capacity added to warehouses and repair facilities Which warehouses and repair facilities to use (binary) When to expand warehouses and repair facilities (binary) 	Minimize cost	Multiple products	Yes, for return volumes only	MINLP with genetic algorithm

Author	Decision variables	Objective function	Product variety	Multi-period support	Classification
Tsai and Hung [39]	<ul style="list-style-type: none"> Product flows to disassembly, scrap, and finished goods Which suppliers to choose (binary) Deviations from goals (environment, efficiency, short-term effectiveness, long-term effectiveness, etc.) 	Minimize deviation from weighted goals	Single product	No	MILP
Pishvae et al. [40]	<ul style="list-style-type: none"> Product flows to various collection centers, recovery centers, and disposal centers Which facilities to use (binary) 	Minimize cost	Single product	No	MILP with simulated annealing
Salema et al. [41]	<ul style="list-style-type: none"> Product flows to various entities Product stocked at various entities Unsatisfied demand at each entity Which entities to open (binary) 	Minimize cost	Multiple products	Yes, macro and micro time scales for demand, return volumes, and transportation costs	MILP
Lee and Dong [42]	<ul style="list-style-type: none"> Product flows to various entities Product returns from various entities Which entities to open (binary) 	Minimize cost	Single product	Yes, for return volumes only	MILP with tabu search
Vasil (see Chapter 4)	<ul style="list-style-type: none"> Product flows to various rework activities and recovery channels Which rework activities and recovery channels to enable 	Maximize profit	Single product	Yes, return volumes, price erosion, and COGS erosion	ILP

2.8 Key Distinctions between This Research and Prior Publications

Guide and Van Wassenhove point to numerous, unresolved, managerially relevant issues that deserve further investigation and inter-industry validation [43]. We attempt to address some of these issues in our research.

Firstly, our ARM model attempts to build on the valuable capabilities of existing work:

- We support sensitivity analysis and ranges of possible values, as Tan and Kumar's model [23], and similar to the probability-bound analysis of Hamza et al. [16].
- We incorporate the returns life cycle, as does Guide et al [25].
- Understanding the importance of price and cost erosion, shifting demand and return rates over time, and other time-based factors, we include a time component in our model like Salema et al. [41], Guide et al [34], and others.
- We support multiple product return conditions, similar to multiple product types.

Secondly, we offer some novel contributions to reverse supply chain modeling:

- We seek to provide a robust model whose formulation can dynamically accommodate a variety of rework options and recovery channels.
- We incorporate setup costs in our go/no-go decisions for enabling rework activities and recovery channels. The preponement model proposed by Guide et al. does not support this [25].
- Instead of using fractional factorial design proposed by Georgiadis et al. [24] to test the robustness of a particular design, we find optimal solutions over the solution space of all possible permutations.
- Our model does not include any parameters that need to be calibrated or estimated, unlike other models.
- Most other models are tactical [23], but use specific time-based product flows to assist strategic decision-making.
- Our model is designed to provide optimal solutions exceedingly fast—within 2 seconds—as a tuned MILP, in comparison to the multi-hour run times and approximately-optimal outcomes of MINLP formulations.

Thirdly, we are excluding some capabilities to keep the problem space tractable and meet other objectives, such as simplicity:

- We do not make facility location decisions.
- We do not examine delays caused by suppliers, transportation, etc., as does Tan and Kumar [23].
- We do not incorporate demand amplification as do Tan and Kumar [23].

Chapter 3. Organizational Assessment

We conducted our research and case studies at Dell, with the assistance of numerous employees. Most of our discussions took place at Dell’s headquarters in Round Rock, TX and neighboring Austin, TX, though we also met with team members in Nashville, TN and Lebanon, TN. We also toured forward fulfillment and returns facilities operated both by Dell and by third parties at these locations.

To give the reader some background in the environment in which we developed the ARM model and related tools, we begin by discussing Dell’s long-term strategy. We then discuss the reverse supply chain groups at Dell using three different perspectives, or lenses: structural, political, and cultural, an approach developed at MIT by Ancona et al. [44]. Guided by these lenses, and with the help of assessment tools from Janse et al. [3], Kumar and Putnam [13], and others, we provide an in-depth assessment of Dell’s current reverse supply chain capabilities. We conclude with some suggestions on how to drive change in this environment, informed from our success in developing and deploying the ARM model.

3.1 Strategy

In the face of its shrinking personal computer market share and expansion into IT services, Chairman and CEO Michael Dell has redefined Dell’s purpose as “delivering technology solutions that enable people everywhere to grow and thrive” [45]. Toward that end, Dell’s Executive Leadership Team (ELT) has rolled out a multi-year strategy based on three pillars, which we summarize in Table 9. The pillars are prominent not only in the company’s long-range vision; as a practical matter, each active initiative at Dell aligns with one of these pillars. All organizational groups are focused on meeting the ELT’s ambitious goals.

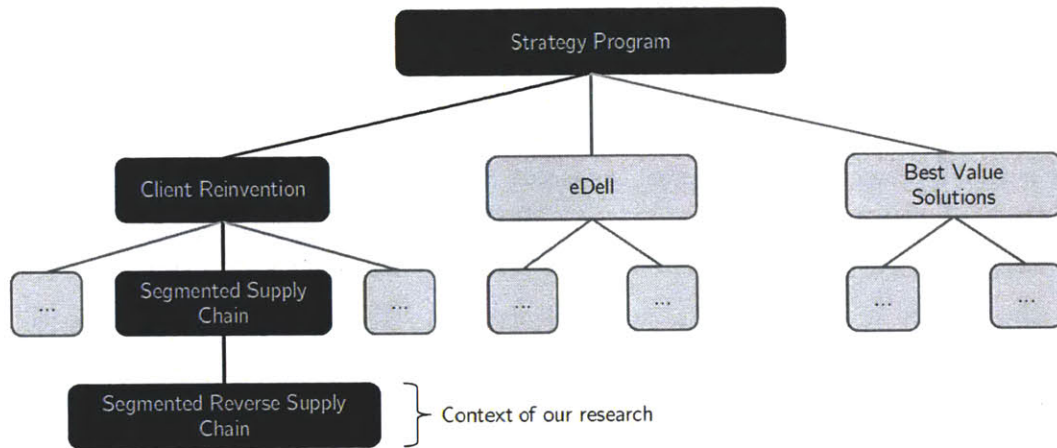
Table 9: Dell’s strategic pillars

Strategic pillar	Explanation
Client reinvention	Optimize operations to align with customers’ needs through fewer configuration options, faster ship times, and cost reductions.
eDell	Modernize online presence with better buying experience and social networking-oriented community support tools.
Best value solutions	Provide fully integrated hardware, software, and service stacks to meet business customers’ unique needs.

Within the client reinvention pillar, Dell defines 11 categories of objectives—called workstreams—one of which is a *segmented supply chain*. This objective is to create multiple supply chains that deliver products in the manner most appropriate for the customer, product, and market. The segments include Dell’s traditional Configure to Order (CTO) model for custom-built systems that require longer lead times, as well as the introduction of a Build to Plan (BP) supply chain for retail orders and a Fast Track supply chain for preconfigured systems held in inventory ready to ship quickly.

We align our research with the segmented supply chain workstream with a goal of a segmented *reverse* supply chain, as shown in Figure 14. With this goal, we raise awareness that segmentation is appropriate not only for the forward supply chain, but for the reverse supply chain as well. Due to a variety of customer needs across products, regions, and markets, the reverse supply chain cannot operate with a one-size-fits-all model. Rather, it must select rework activities and prioritize recovery channels carefully by understanding these unique needs. Our work on the ARM model helps Dell do this.

Figure 14: Context of our research within Dell's strategic initiatives



3.2. Three Lens Analysis

Rather than attempting a broad three-lens analysis of a global, 96,000-employee corporation, we focus our analysis on the groups who share responsibilities for Dell's reverse supply chain: Asset Recovery Business (ARB), Global Service Providers (GSP), Global Reverse Logistics (GRL), and Dell Financial Services (DFS). While ARB, GSP, and GRL employees served on our working group, due to its independence, DFS employees did not. Established as a subsidiary structured within Dell's financial arm, DFS leases all types of Dell- and non-Dell branded IT equipment to consumer, SMB, and public segments; it also provides financing for purchases.

We summarize the reverse supply chain responsibilities of each of these groups in Table 10.

Table 10: Reverse supply chain responsibilities across Dell's groups

Group	Reverse supply chain responsibilities
ARB (U.S. and EMEA only)	<ul style="list-style-type: none"> • Manages the online store for refurbished systems, the Dell Outlet (http://www.dell.com/outlet) • Manages the online store for refurbished parts • Sells systems, peripherals, parts, and scrap through auctions, bulk deals, and 3PLs including GENCO's marketplace (http://www.gencomarketplace.com)
GSP	<ul style="list-style-type: none"> • Issues CRAs to customers for returns and providing shipping labels • Processes returns for some products, like printers and LCDs • Handles all warranty claims, overseeing warranty servicing with the help of third parties • Returns defective parts to suppliers for crediting • Manages the inventory to support warranty claims
GRL	<ul style="list-style-type: none"> • Handles logistics of most returns • Oversees 3PLs receiving and processing returns • Makes teardown and safety capture routing decisions • Performs ARB functions in APJ
DFS	<ul style="list-style-type: none"> • Performs ARB, GSP, and GRL functions on DFS equipment • Manages the online store for refurbished DFS equipment (http://www.dfsglobalsales.com) • Manages an online auction marketplace for refurbished DFS equipment (http://www.dellauction.com)

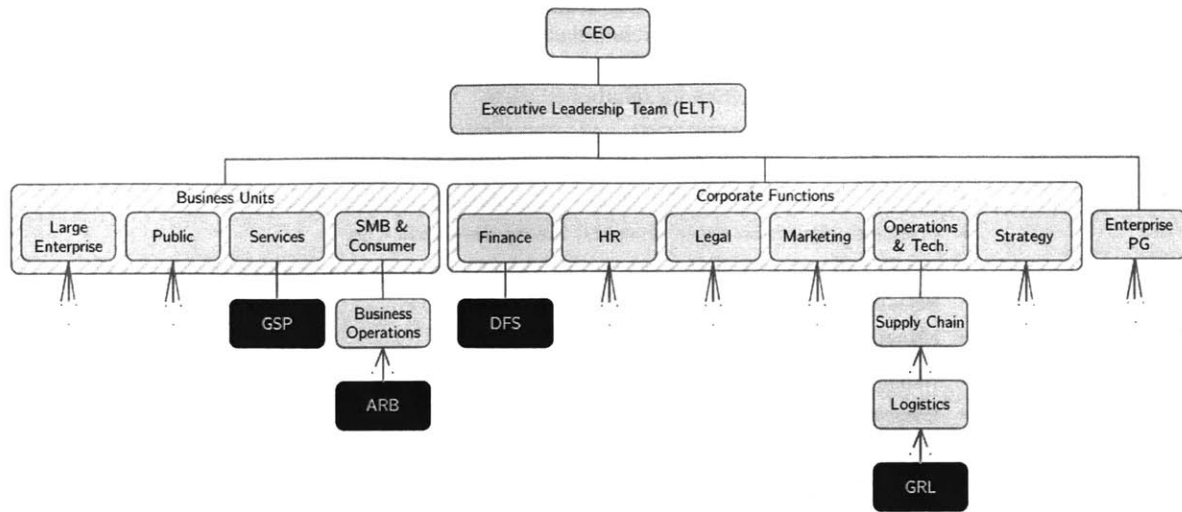
3.2.1. Structural Analysis

Structurally, Dell arranges itself into 11 branches: four business units (BUs), six corporate functions, and the Enterprise Product Group. The BUs are market-centric structures focused on global and national corporate customers (Large Enterprise); educational institutions, government, health care, and law enforcement agencies (Public); organizations with consulting and cloud computing needs (Services); and small and medium businesses and end-users (SMB & Consumer). The corporate functions provide financial, HR, legal, marketing, operations and technology, and strategy support to these BUs. The Enterprise Product Group designs, builds, and markets Dell's enterprise products. The branch heads, as executive officers of the company, serves on the ELT alongside CEO Michael Dell [46].

3.2.1.1. Reverse Supply Chain Functions

As shown in Figure 15, Dell locates its four groups with principal responsibility over its global reverse supply chain functions spread in four different branches of the organization: ARB sits within the SMB & Consumer BU, GSP within the Services BU, GRL within the Operations & Technology corporate function; and DFS within the Finance BU. Despite their locations within the organizational structure, ARB and GSP service all of Dell's BUs, not just their respective BUs.

Figure 15: Reverse supply chain functions within the structural organization of Dell



The division of reverse supply chain responsibilities among the branches has at least several important ramifications:

1. The four groups must communicate cross-functionally to implement some changes or new capabilities. Currently Dell has no general-purpose standing meetings among these groups, though ad-hoc meetings do occur frequently.
2. Identifying and sharing best practices may be difficult, as groups do not have detailed visibility into each other's' processes.
3. Without economies of scope, inefficiencies may arise as groups duplicate processes for their particular domains. For example, DFS has its own contracts and storefronts to refurbish and resell its equipment.
4. Over time, a patchwork of seemingly arbitrary or ambiguous responsibilities may develop. For example, for historical reasons ARB happens to manage the contract for one of GSP's 3PLs. In the worst case, this may lead to improper oversight of the 3PL.
5. Without a single point person for reverse supply chain issues on the ELT, strategic improvements therein—such as design for disassembly (DFD)—require additional coordination among its members.

In a further division of responsibilities, Dell outsources many of its reverse supply chain functions to third parties, including GENCO, CEVA, Decision One, CTS, and Flextronics. While many of these 3PLs are global, Dell does not always leverage their global footprint. For example, while Flextronics manages the refurbishment process for computer systems in EMEA and APJ, GENCO principally provides this service in the U.S. Consequently, global changes or new capabilities sometimes require coordi-

nation across multiple 3PLs, and any associated software system changes are not necessarily one IT project, but several.

3.2.1.2 Regional Factors

While Dell's BUs and corporate functions have directors and managers with global roles, many teams still organized regionally. There is strong precedent for this; before customer-focused BUs, Dell organized its functions along regional lines. Based on its regional focus, and natural differences among regions including return rates, labor rates, logistics, and the dynamics of secondary markets, the overall arrangement of the reverse supply chain varies by region. We summarize these differences in Table 11.

Table 11: Regional differences in Dell's reverse supply chain

Region	Return rates	Primary 3PLs	Primary RL functions
DAO	10-20% in North America; lower elsewhere	<ul style="list-style-type: none"> • GENCO in U.S. and Canada • Decision One • Additional third parties in Latin America 	<ul style="list-style-type: none"> • Full suite of repair and refurbishing capabilities, including teardown for parts
EMEA	Less than 5%	<ul style="list-style-type: none"> • Flextronics 	<ul style="list-style-type: none"> • Repair and refurbish from all regions • Systems localized for resale in UK only
APJ	Less than 5%	<ul style="list-style-type: none"> • Flextronics (transitioned in 2010 from Dell's in-house capabilities) 	<ul style="list-style-type: none"> • Repair, refurbish, and resell in one of six countries closest to country of origin • Systems localized for simplified Chinese or UK English • Specific policies differ due to variations in labor costs and logistics costs

Within ARB, GSP, DFS, and GRL, regional players meet as a group to review metrics and discuss ongoing projects in their respective regions. GRL leaders from DAO, EMEA, and APJ meet monthly, for example.

3.2.1.3. ARB Accounting

Dell's accounting office maintains a shadow profit and loss (P&L) statement for the Asset Recovery Business for tracking and compensation purposes. It does not report this P&L publicly, however; instead, it spreads ARB's income and expenses to other P&Ls that the company does disclose: the BU P&Ls. How it does so is rather complex. Roughly:

- The BU responsible for authorizing the return sees a chargeback for the return. The specific chargeback is a fixed percentage of the standard cost of the item at the time of the return.
- The BU responsible for the ultimate sale of any asset from the reverse supply chain, regardless of the originating BU or condition of the item, recognizes the revenue of the sale, less any transformational costs such as repair.

- ARB's fixed costs and overhead are not assigned to any particular BU, but rather recognized as selling, general, and administrative (SG&A) overhead on financial filings.

Given these intertwined financials, the BU heads and ARB manager collaborate on the decisions affecting their respective business. Specifically, for refurbished inventory sold through their respective sales channels the BUs have a say in not only price but also the products' bill of materials (BOMs). When it comes to BOMs, the BUs require ARB's refurbished products match the specifications of their new counterparts. For example, a laptop with 2 GB of memory made by a particular supplier cannot be sold with less memory, or from another supplier, unless that supplier's part is on the BU's list of compatible substitutes.

The BUs' requirements on ARB may be ideal for the BU, since they promote brand continuity and component compatibility, but not necessarily optimal for ARB, which seeks to maximize net recovery. The BUs' policies add to transformational cost without, necessarily, a proportional boost to resale revenue. ARB is exploring alternative, less-restrictive recovery channels through its 3PLs relationships as it explores ways to maximize net recovery.

Another ramification of the financial reporting structure is that the BUs do not share ARB's interest in net recovery. They concern themselves with recovery revenue, which ignores transformational costs, as only recovery revenue integrated with the BUs P&Ls; the costs are reported elsewhere. (The difference between *net recovery* and *recovery revenue* are illustrated in Figure 3 on 22.) This distinction is important, since BUs do not take a hit for any inventory holding costs (rolled into transformational costs), meaning they prefer ARB hold inventory in anticipation of higher margin-sales. This locally-optimal decision-making results in a less-than-optimal outcome for Dell. Inventory holding costs should be an important factor in their pricing decisions, but they are not incented to consider these costs today.

3.2.1.4. SFF CLSC Planning

For the majority of 2010, Dell managed its SFF portfolio through a Communications Services BU. The ELT has since disbanded the unit, reasoning that its functions were better placed across the other BUs, who manage research, design, sales, marketing, and customer services for their respective target markets. With or without the BU, however, Dell is being challenged to pull together an optimized reverse supply chain for SFF devices.

Program managers assigned to GRL have been leading the efforts to coordinate the SFF reverse supply chain across GRL, GSP, and ARB groups. Since Dell has not introduced a new form factor in some time, no standard process is in place to ensure the decisions progress smoothly and linearly. For one, it meant that these stakeholders were not involved early in the product planning process, which has led to additional challenges:

- The design teams did not engineer the products with the reverse supply chain in mind, i.e. designing for disassembly (DFD). Scanning barcodes upon receipt of a return requires removing the battery and other components, which is a labor-intensive process.

- Contracts with the telecommunications companies were generous with their return policies, leaving the door open for a large number of returns, especially E&O returns.
- Contracts with the SFF original design manufacturers (ODMs), including FIH and Qisda, allowed Dell to exchange DOA devices for like-new devices instead of receiving credit for them, which adds logistics complexity to the reverse supply chain.

The small form factor itself poses unique challenges in the reverse supply chain that require careful consideration. There are distinctive repair capabilities and requirements, such as radiofrequency (RF) testing; tie-ins to telecommunications companies for service; special hardware, including international mobile equipment identity (IMEI) numbers and subscriber identity module (SIM) cards for telephony identification; and the unique revenue model based in part on service activation bounties. With limited resources and no incentive for any of the organizations to take ownership of the process, seemingly simple decisions such as where returns are to be processed or the extent to which warranty stock are to be replenished with refurbished product returns require extensive negotiations among groups.

3.2.2 Political Analysis

Dell relies on thousands of IT systems to run its online storefronts, transact orders, ensure trade compliance, manage inventory, and handle its reverse supply chain routing decisions, among many other functions. While many of these systems have a queue of enhancement requests stemming from business needs, Dell caps its spending on internal IT to 1.74% of revenue [47]—regardless of the potential return on investment (ROI)—and uses a process called the Approved Global Operations Process (AGOP) to filter and prioritize these requests. This limit is lean compared with the 2-4% cap set by Dell’s competitors such as HP [48]. It then manages approved projects through a phase-gate process that varies by branch of the organization: a custom phase review process (PrP) in Operations & Technology, for example, and the common software development life cycle (SDLC) process within the IT organization. To get a program approved in this resource-constrained environment, program managers must demonstrate a solid business case. The most successful of them are also strong advocates and negotiators on behalf of the teams they represent.

With respect to the development of the ARM model, rather than add our project to an IT prioritization queue, we opted for a path requiring no IT resources for development or integration. We based this decision, in part, on Lin’s difficulties with integrating his SORT tool into the IT infrastructure [29]. Instead of purchasing specialized software, we implemented the model using existing tools with which our stakeholders are familiar. While this option is not available or practical in all cases, in our case it turned out well. We spared Dell additional costs while delivering a more familiar and approachable solution.

In terms of collaboration, we found that the stakeholders came together to identify common goals quickly and naturally, even across functional groups and regions. The exchange of ideas between ARB and GRL members was especially smooth. We cite their mutual interests in maximizing net recovery as a possible motivating factor.

3.2.3. Cultural Analysis

Employees, especially the ones with a tenure of five years or more, speak proudly of the “speed of Dell,” a nod to Dell’s fast-paced atmosphere where decisions happen not only quickly, but at a faster rate than at competitors. Decisions came especially quickly throughout its phenomenal growth in the 1990s, driven by a brilliant supply chain design that employed a direct model instead of holding inventory [48]. Today, with the increasing commoditization of personal computing hardware, Dell’s segmented supply chain approach now accommodates finished goods inventory in certain cases. Further, Dell has identified a need to standardize, consolidate, and harmonize many of the IT systems that grew up separately and independently across regions. The net effect is a slower speed for Dell, with more methodical, time-consuming decision-making.

One unique aspect of Dell’s office environment is its “one-on-one” culture. Seemingly everyone in the company embraces the ability to meet with anyone else, regardless of organization or title, through a simple request. Many productive meetings occur in these one-on-one settings. In fact, experienced Dell employees typically use one-on-one meetings as the bookends for group meetings. That is, before and after larger meetings they will meet individually with stakeholders to answer questions, identify concerns, and perform joint problem solving. We also employed this approach with our working group members after discovering group meetings, on their own, were not sufficient to make progress. For new employees or those of a lower rank, one-on-ones provide great exposure and an excellent way to share new ideas. Dell prides itself on being a meritocracy, and one-on-one meetings not only attest to that belief, but also are likely responsible for it.

We observe ARB and GRL groups operating somewhat within their own ecosystem, or subculture. Processing returns, in general, is not a glamorous business; at least not as glamorous as, say, launching a line of smartphones or opening a cloud-computing datacenter. Further, their groups’ contribution to the bottom line is at most 2%. Consequently, from above they do not receive much visibility or IT resources, or implement some of the more rigorous analytical processes for which Dell is well-known. Our work with the ARM model attempts to introduce some additional processes to the group to make collaboration easier and the groups more efficient.

3.3 Reverse Supply Chain Maturity Assessment

From its origins as a physical Dell Outlet storefront in the back of a factory in Round Rock, TX, Dell’s reverse supply chain has matured immensely. In the following subsections we describe its current state in terms of its structure, collection methods, capabilities, and return minimization initiatives. We then provide a description of a specific returns facility in Lebanon, TN where we tie these observations together.

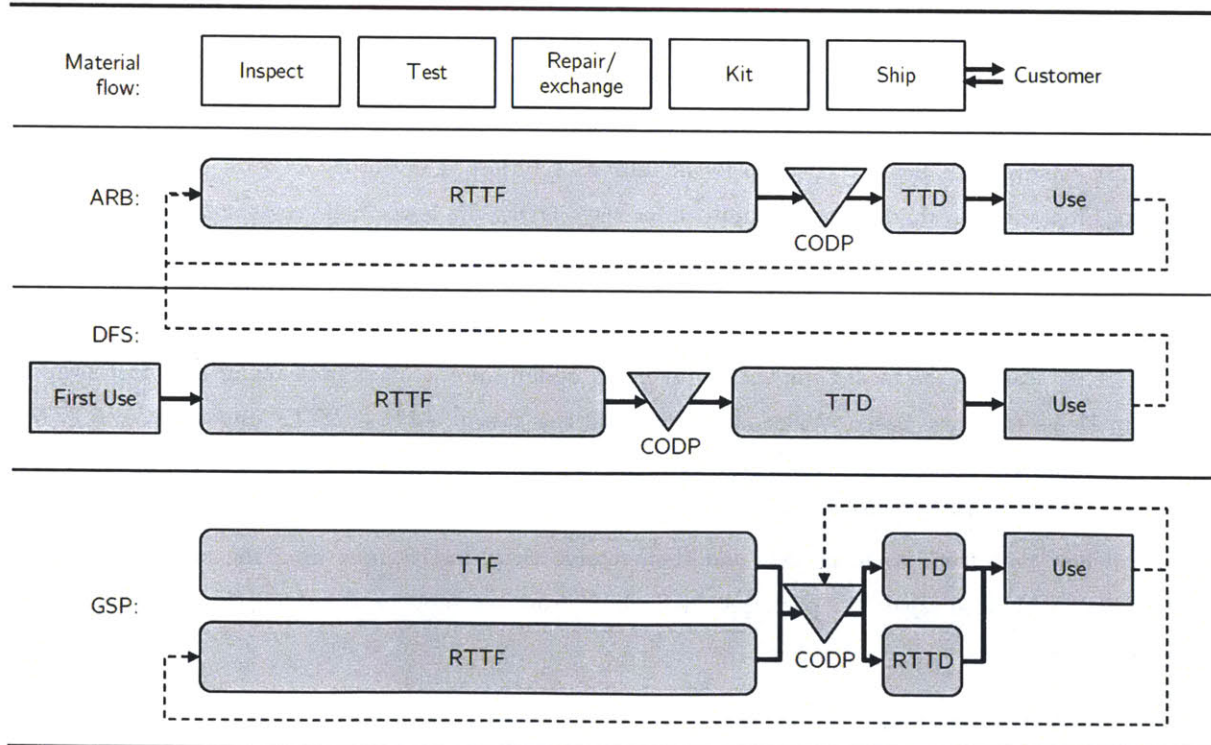
3.3.1. Structural Framework

Wikner and Tang's provide a structural framework for designing and evaluating closed-loop supply chains [4]. They identify nine unique types of CLSCs that share the following five building blocks, which we review in Table 12.

Table 12: CLSC building blocks described by Wikner and Tang [4]

Acronym	Definition	Explanation
CODP	Customer order decoupling point	The point at which the "pull" supply line is transformed into a "push" supply line due to a customer order. At the CODP is typically materials of some intermediate form that are transformed to a final product on the basis of a customer order. Researchers also refer to the CODP as the "decision point" or the "order penetration point."
TTF	Transform to forecast	The segment of a supply chain where materials are processes, or transformed, in some way based on a forecast, thus occurring before the CODP.
TTD	Transform to demand	The segment of a supply chain where materials are processes, or transformed, in some way based on actual demand, thus occurring after the CODP.
RTTF	Retransform to forecast	Similar to TTF, but specifically refers to the reverse supply chain where previously shipped materials are recovered and transformed for a second time.
RTTD	Retransform to demand	Similar to TTD, but specifically refers to the case where a customer receives their specific part back in a refurbished state

Figure 16: Reverse supply chain strategies of Dell businesses using Wikner and Tang's structural framework [4]



We find this framework useful in the supply chain designs employed by ARB, DFS, and GSP's warranty program, where they swap out customers' non-working parts for working ones. ARB and DFS both employ a *RTTF and TTD* design, while GSP employs a more complex *TTF + RTTF and TTD + RTTD* design that makes use of more parts inventory. We illustrate these three designs in Figure 16.

3.3.1.1. ARB Supply Chain Design

ARB follows a closed-loop *RTTF and TTD* design, which Wilker and Tang denote as "Configuration IV." In its normal process, ARB collects computer systems such as laptops and desktops, refurbishes them, and kits them—a process that includes a wipe down (cleaning); merging with appropriate software and peripherals (S&P) including CDs, keyboards, and mice; boxing; and storing as finished goods inventory. Once in finished goods, ARB makes the systems available for sale on the Dell Outlet website, or through other channels. The online point of sale provides the CODP, after which ARB retrieves the desired products from finished goods inventory, affixes shipping labels, and ships the units to the customers. The design is a closed-loop since customers' returns re-enter the process. Within their lifetime, some Dell systems loop through the process multiple times.

3.3.1.2. DFS Supply Chain Design

DFS employs the same *RTTF and TTD* design as ARB, however its P/D ratio is lower. The P/D ratio reflects the proportional size of the forecast-driven side of the supply chain to the demand driven side, where P is production lead time and D is delivery/fulfillment lead time; essentially, these are the lengths of the supply chain before and after the CODP point, respectively. Given similar designs, its lower P/D ratio means DFS is exposed to lower inventory holding costs, fixed costs, and E&O risk than ARB. It achieves this lower P/D ratio by holding its refurbished inventory as work in progress (WIP), and not proceeding to the kitting step until after the CODP. Its lower fixed costs stem from its smaller footprints: unkitted systems occupy about a quarter of the space as their kitted counterparts.

3.3.1.3. GSP Supply Chain Design

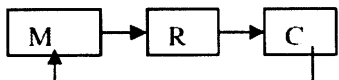
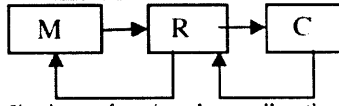
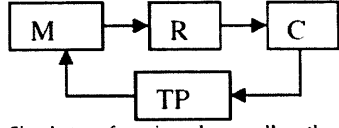
GSP features the most complex supply chain design since it offers both repair and part swap services to Dell's customer base. Wilker and Tang call this design *TTF + RTTF and TTD + RTTD*, or "Configuration IX." In the case of repair, GSP stocks an inventory of parts and will deploy a technician to the customer at the CODP to perform the repair work. In the case of a swap, GSP will send the customer a new part from its inventory, and then receive the defective part from the customer. At that point, it may elect to repair the part, get credit for it from the supplier if covered by a warranty, or dispose of properly.

3.3.2 Collection Methods

Savaskan et al. describe the principal collection methods for remanufacturing in a reverse supply chain as shown in Figure 17: back to the manufacturer directly, via a retailer, or via a third party. Dell

uses hybrid collection methods, which is not surprising considering it uses both direct and retail sales channels in its forward supply chain. We illustrate this hybrid model in Figure 18. Here, *S* represents Dell's suppliers, which produce motherboards, hard drives, and other computer parts. *M* represents Dell's OEM or ODM manufacturer Dell's computers and SFF devices, such as Foxconn and Qisda. *R* represents retailers that sell Dell products, including Walmart, Staples, and Best Buy. *C* represents the ultimate customer, a consumer, SMB, or other organization. *TP* represents the 3PL provider processing the returns, such as GENCO, Flextronics, or DecisionOne. Finally, *TP2* represents another third party, who buys refurbished products from the *TP* through a revenue share, auction, or special arrangement with ARB, GSP, or DFS.

Figure 17: Collection methods described by Savaskan et al. [49], adapted by Kumar and Putnam [13]

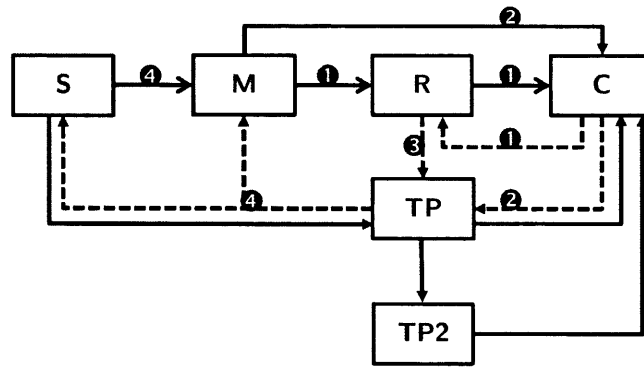
Collection method	Explanation M = manufacturer R = retailer TP = third party C = consumer	Examples of products
Manufacturer collects from consumer. Retailer is not involved	 <pre> graph LR M[M] --> R[R] R --> C[C] C --> M </pre>	<ul style="list-style-type: none"> • Xerox and Canon use prepaid mail boxes • Hewlett Packard picks up from local offices
Retailer collects from consumer and manufacturer buys-back from retailer	 <pre> graph LR M[M] --> R[R] R --> C[C] C --> R R --> M </pre> <p>Simple transfer price schemes allow the manufacturer to coordinate the supply chain</p>	<ul style="list-style-type: none"> • Kodak single-use cameras are returned to a retailer for developing and Kodak "buys-back" • Refrigerators & televisions are traded-in
Third party collects used products from the consumer	 <pre> graph LR M[M] --> R[R] R --> C[C] C --> TP[TP] TP --> M </pre> <p>Simple transfer price schemes allow the manufacturer to coordinate the supply chain</p>	Old cars (Ford, GM, Chrysler, BMW, Fiat and Renault) dealerships, junkyards, recycling centers and disassemblers sell recovered parts or materials back to manufacturer

We show five distinct collection methods in the figure:

1. Retail sales. When a customer purchases a Dell product through a retailer, he or she may return it to the retailer under the retailer's return policy. This policy differs from Dell's own return policy, which gives retailers a specific number of days to return products. Due to the discrepancy, the retailer may or may not be allowed to send the return back to Dell for full credit. The retailer may have to accept partial credit or find a recovery channel depending on the age of the product. For those returns Dell is willing to accept, the retailer bundles them and sends them to the TP working on behalf of Dell.
2. Direct sales. When a customer purchases a product from Dell directly, he or she must call GSP and obtain a CRA to send the item back for credit. Depending on when the customer purchased the item, he or she may not be eligible for a CRA. With a CRA, a customer receives a shipping label to send the product to the TP.

3. Retail E&O. Retailers may return unsold, excess inventory to Dell under the same terms as products they sell to customers.
4. Defective products and parts under warranty. As it discovers defective products and parts among returned inventory, Dell checks to see whether any of it is covered by suppliers' or manufacturers' warranties. When this is the case, Dell sends back these products and parts for credit, or for exchange with new or refurbished replacements, depending on the contract.
5. EOL (combination of methods 1, 2, and 3). Dell accepts end-of-life products from consumers and retailers, of which it properly disposes in accordance with its environmental policies and government regulations. It accepts these returns at no cost.

Figure 18: Dell's collection methods



Strikingly, Dell as an entity is not represented by any of the boxes in Figure 18. Instead, *S*, *M*, *R*, and *TP* are all third parties operating on its behalf. Its involvement in collecting returns is limited to issuing the CRAs, managing the IT systems to authorize and track returns, and overseeing the 3PLs that implement the reverse supply chain. This arrangement makes it especially challenging for Dell to identify the root causes of the return volume, and to close the loop by sharing this information with the design and marketing groups. We discuss this in more detail in the next subsection.

3.3.3. Initiatives to Minimize Returns

Dell has direct visibility into returns stemming from direct sales. With each CRA it issues, it associates a reason code. Common reasons include a customer changing his or her mind, or confusion regarding how to use the device, or a discrepancy between actual and expected capabilities. An ARB team data mines these reason codes periodically to detect trends and resolve problems.

In the realm of retail returns, Dell has less visibility. Often, retailers do not ask customers for the reason they return Dell products. When they do, the information typically is not recorded or shared with Dell. The reasons are important, however, especially since retail return rates are historically higher than

return rates from direct sales. The ARB team contacts random samples of retail customers to gauge their satisfaction with purchases. Interestingly, customers who receive these types of calls are actually much less likely to return their products. We speculate the reasons for this are two-fold: customers have an easy way to get common questions answered, which, without such a channel, might otherwise be prone to give up and return the product; and, second, the attention they receive from the call boosts the goodwill they feel toward Dell and its brand.

The calls provide other benefits for Dell, since they reveal problems that would otherwise go unnoticed. For example, Dell recently discovered a problem with its packaging. It sells some lines of laptops and desktops with a big picture of a black, gray or white machine on the front, with the actual color of the system indicated by a small picture on the side of the box. As it turns out, one of the common reasons for returns of these products is confusion over the actual color. Customers buy what they think is a black product, for example, only to open it and find a red one. They then go back to the retailer to swap it with the correct color. Dell, unfortunately, incurs a logistics and refurbishment expense, since the (unused) black machine is nonetheless sent through the triage and refurbishment process of its reverse supply chain. By identifying a packaging problem and sending that information to the marketing team, ARB is playing an important role in minimizing returns.

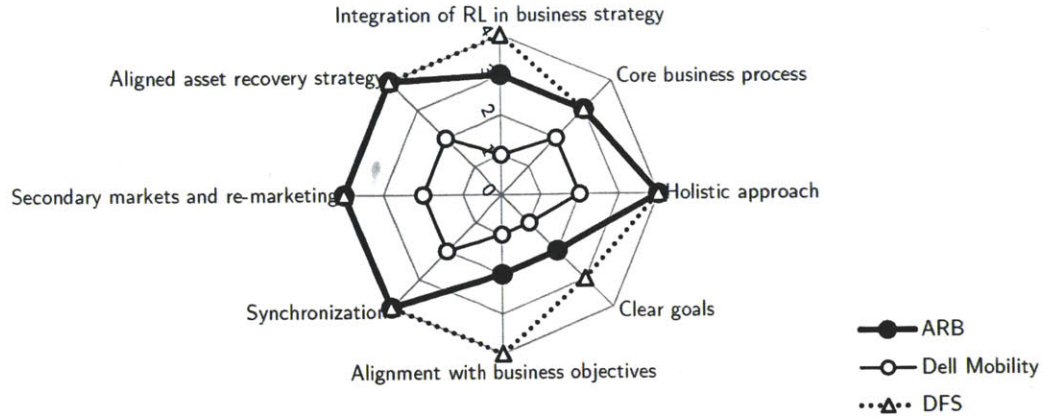
3.3.4. Capabilities Assessment

So far, we have explored the structure, collection methods, and initiatives within Dell's reverse supply chain. As Janse et al. note, this alone does not provide a comprehensive picture as it does not capture the firm's philosophy of returns and degree to which reverse supply chain thinking permeates core businesses. To fill this gap, they provide a framework to assess these capabilities, defining eight dimensions along which a firm may evaluate its reverse supply chain [3]:

1. Integration of reverse supply chain management in supply chain strategy,
2. Managing reverse logistics as a core business process, using a holistic supply chain approach,
3. Defining clear reverse logistics goals with respect to the end-to-end process,
4. Alignment with business objectives,
5. Synchronization in spare parts management,
6. Knowledge of secondary markets,
7. Remanufacturing capabilities, and
8. An aligned asset recovery strategy.

In Figure 19, we use this framework show our judgment of the relative strengths and weaknesses of ARB (including GRL), Dell Mobility, and DFS. For each capability, we offer ratings from one (immature) to four (mature), as suggested by Janse et al.

Figure 19: Assessment of Dell's capabilities in the reverse supply chain using Janse et al.'s eight-part framework [3]



For the purposes of our evaluation, we consider Dell Mobility as the combined efforts of ARB and GSP to handle SFF returns. Including Dell Mobility adds an interesting dimension to the analysis, as Dell is still in the process of designing and ramping up reverse supply chain capabilities for SFF devices. From visual inspection, we observe that DFS provides the most mature reverse supply chain capabilities, followed closely by ARB, then with Dell Mobility trailing.

We present the overall ratings of these three groups in Table 13, noting each group's lowest-rated capabilities. This is the primary purpose of the assessment: to drive improvement projects that improve these capabilities. Overall, we see excellent capabilities at ARB and DFS; their rankings are very high, which aligns with Dell's own beliefs that its reverse supply chain is best-in-class. At the same time, we observe the outlier of Dell Mobility. As a new product line, the capabilities of this reverse supply chain do not yet match those of Dell's longstanding products. There is reason to believe, however, that as this product line matures its reverse supply chain capabilities will match those of ARB overall.

Table 13: Summary of Dell's capabilities in the reverse supply chain

Entity	Score	Weakest capabilities
ARB	3.3 ★★★★★	<ul style="list-style-type: none"> Clear reverse logistics goals for end-to-end processes Alignment with business objectives
Dell Mobility	1.6 ★★★★★	<ul style="list-style-type: none"> Integration of RL in business strategy Clear reverse logistics goals for end-to-end processes Alignment with business objectives
DFS	3.8 ★★★★★	<ul style="list-style-type: none"> Managing reverse logistics as a core business process Clear reverse logistics goals for end-to-end processes

We assess ARB as performing very well in remanufacturing assets, assessing secondary markets, and recovering value. As we highlight later, in Chapter 8, ARB employs a sophisticated teardown model for harvesting parts, which it uses for various recovery channels, including remanufacturing. ARB scores

highly in the corresponding capability of “synchronization.” ARB’s weakest areas relate to goals and alignment. While ARB has goals in place for virtually all processes in the reverse supply chain, it does not work aggressively to meet these goals, especially in WIP, finished goods, and cost-per-box metrics. Additionally, it adapts to business objectives rather than being aligned with them. For example, it may change the price of availability of products on the Outlet store in reaction to short-running promotions in the retail channel, but coordination does not happen in the opposite direction, which would allow segments to drive sales to ARB when they experience high volumes of certain products.

Dell Mobility presents a challenge for Dell. Despite the maturity of ARB, the capabilities of the Dell Mobility reverse supply chain are comparatively weak. Reverse supply chain capabilities are not considered the SFF product design, marketing, and sales teams; rather, these capabilities are built on as addendums to their strategies. As a result, goals and alignment are not yet in place.

DFS offers the most mature model. Its business model is designed with the reverse supply chain in mind. Its off-lease returns come back to DFS already fully paid for, so its reverse logistics operation is a profit-generating segment of the business. It attempts to strike a balance between degree of repair and price so as to maximize profit, however one of its weaker areas—limited goals in managing the end-to-end-process—means it may not yet have found this balance. It may have additional exploitable value to squeeze out of its assets, and may be able to take a cue from ARB to learn how to do this.

The overall picture we see is that, even in a company with very mature reverse supply chains, there is at least one product segment (Dell Mobility, or SFF) where the process is not yet mature. With different capabilities expressed within the same firm, and, in fact, underneath the same roof, the groups should seize the opportunity to codify and communicate their best practices in order to leverage each other’s capabilities. In this manner, future product launches, especially those that present new technologies or form factors, can ramp up faster without the growing pains of an immature reverse supply chain.

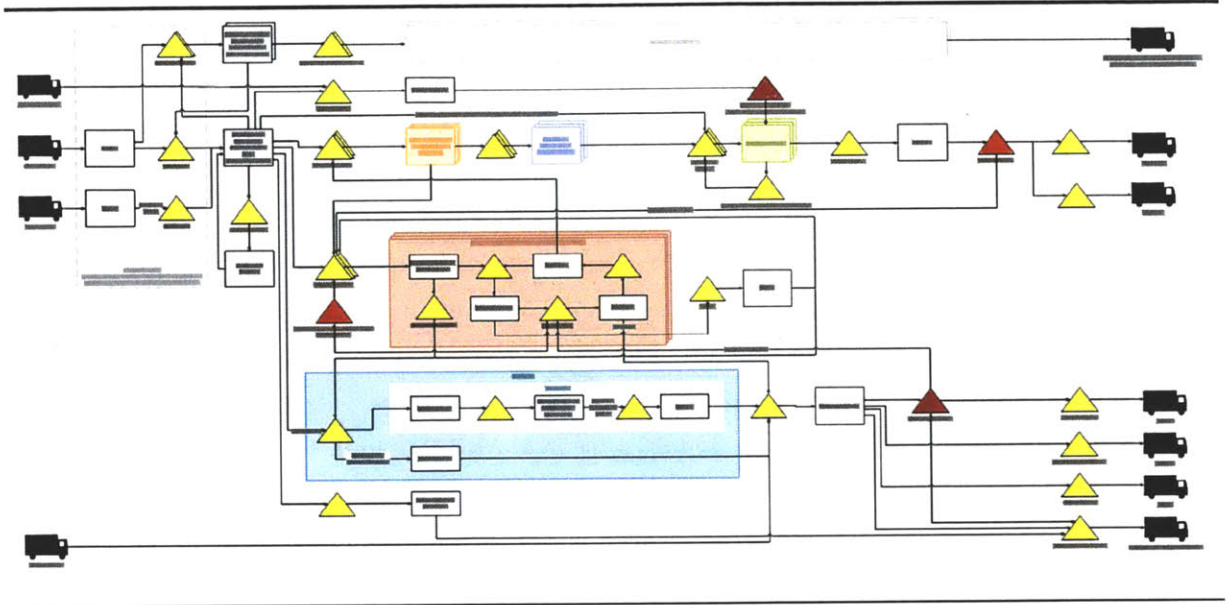
3.3.5. Case Study: GENCO Facility

To conclude our assessment of Dell’s reverse supply chain, we offer a brief look at the processes at one of Dell’s 3PL partners, GENCO. GENCO operates a number of facilities for Dell, including one in Lebanon, TN. In Figure 20, we present a high-level overview of the main process flows GENCO manages on behalf of Dell for ARB. (In separate space on the factory floor, it also manages the returns process for DFS.) To protect Dell’s proprietary information, we mask the details. Even so, the robustness of the processes is clear. Refined with years of experience, this return facility processes multiple types of returns for Dell:

1. Dell-branded peripherals, including printers and monitors
2. Non-Dell branded peripherals, including TVs, gaming consoles, and cameras
3. Dell-branded laptops and desktops
4. SFF devices

In the figure, which we base solely on our own observations, we denote temporary staging locations with yellow triangles and long-term inventory locations of AWP (i.e. systems awaiting parts), finished goods, and parts inventory with red triangles. The pink-colored rectangular region in the center of the diagram is where GENCO performs electromechanical repair (EMR), and the cyan-colored rectangular region below it is where GENCO's subcontractor IntelliSol dismantles systems for part recovery.

Figure 20: High-level process flow diagram at GENCO returns processing facility in Lebanon, TN



GENCO provides complete refurbishment capabilities in-house for laptops and desktops. The other types of returns are sold as-is through GENCO's own marketplace under a revenue share arrangement with Dell, or cross-docked to other 3PLs under contract with Dell. For laptop and desktop repair, GENCO follows this standard process "three Rs" process:

1. Receipt:
 - a. Sort: GENCO identifies the return as a laptop or desktop from a retailer or consumer, and sorts it separately from other types of returns, subdividing it by form factor—laptop, desktop, or LFF—which are processed through separate lines optimized for device size.
 - b. Scan and strip: Through integration with Dell's IT system, GENCO scans the service tag of the system, verifies the CRA, credits the customer, and strips peripherals, including power supplies and product CDs, which are stored separately.
 - c. Test: GENCO runs a series of tests to determine any hardware or software problems. The test result of most systems indicates no fault found (NFF).

2. Rework:

- a. EMR: For systems with faults, technicians remove faulty parts, put in new or refurbished replacement parts from a parts supermarket, and re-test. If further failures are found, the repair process repeats.
- b. Burn: The hard drive(s) are wiped of all customer data, and a new machine image, complete with operating system and additional factory-set software, is installed.
- c. Re-kit/pack: GENCO marries systems with their peripherals, wipes everything down, packages them, and stores them in finished goods inventory for sale.

3. Recovery:

- a. Pick: Upon notification from Dell of an Outlet order, GENCO retrieves the desired system from finished goods inventory
- b. Ship: GENCO prints the appropriate label on the box and ships using the designated carrier.

Of course, not all returns follow this “happy path.” Alternative paths include safety captures, a research area for unauthorized or unidentifiable returns, and a teardown area where systems are broken down into reusable parts.

3.4. Driving Change: Techniques for Success at Dell

Each company has unique structural, political, and cultural context. Having reviewed Dell from these lenses, we conclude this chapter with eight techniques for success that we find work well in Dell’s context. We discovered these techniques through some trial and error, as well as in depth discussions with employees. We relied on them while developing the models and processes discussed in subsequent chapters.

3.4.1. Thoroughly Identify Stakeholders

We recommend surveying Dell through strategic, cultural, and political lenses to identify stakeholders. In our case, it was easy to start by choosing organizational units, such as ARB and GRL, and leverage Dell employees’ amenability to one-on-one meetings to introduce ourselves to a broad cross-section of its members. Once we spotted those whose interests or responsibilities intersected one of our projects, we added them to our stakeholder list. We also asked everyone with whom we spoke about others we should meet who might have an interest in our initiatives. In so doing, we were able to pull on a thread that allowed us to identify relevant stakeholders quickly, even without much advance knowledge the relevant organizations. Keep the stakeholder list open as you learn more about the organization and its employees. Our stakeholder list continued to grow as our projects progressed.

3.4.2. Form a Core Team

A core team is an excellent way to give an initiative some structure. At Dell, the concept of a “core team” represents a cross-functional group of stakeholders who work together on a project. Typically a program manager drives the initiative, and the stakeholders previously identified each contribute to the project according to a mutually agreeable project schedule. Often, these core teams spawn from programs sanctioned through the AGOP process according to a phase-gated PrP plan, however this need not be the case. We found a core team added structure and legitimacy to our initiatives even though they operated outside the PrP process. Depending on the project, we set recurring meetings of varying frequency to maintain the momentum of our projects.

Of course, core team meetings come with their challenges. With stakeholders typically distributed globally, meeting times are limited so as to accommodate the disparate time zones. For those working at or near headquarters in Round Rock, TX (Central Time), global meeting times are typically 7 am or 9 pm. Videoconferencing support is quite limited, so virtually all meetings are conducted via teleconference. We find soliciting feedback and fostering fruitful discussions harder to foster through this medium. Further, Dell’s recent initiatives to re-classify a sizable fraction of its Austin, TX workforce as “digital nomads,” who typically work from a home office and journey to campus relatively irregularly and infrequently, make it hard to pull together stakeholders even when they happen to be in the same location. To mitigate these problems, leverage one-on-ones, as discussed in the next subsection.

3.4.3 Conduct Frequent One-on-One Meetings with Stakeholders

The venue where most of a project’s heavy lifting takes place is one-on-one meetings with stakeholders. These meetings present a great opportunity to draw out a stakeholder’s expertise and engage in the type of detailed discussion generally not possible given the time constraints and dynamics of larger-sized meetings. Some employees at Dell find it helpful to set up biweekly or monthly recurring one-on-one meetings with their most important contacts. While managing our projects, we found it helpful to meet as frequently as weekly with our most important stakeholders. Sometimes planning an agenda in advance helped us make productive use of our one-on-one time, although at other times we found open table discussions to be more effective at engaging stakeholders and bringing out creativity.

3.4.4. Identify and Address Key Pain Points

Each stakeholder has a unique list of pain points—or high-priority issues—and we found it useful to keep these in mind. By remembering what is most important to each stakeholder, we were able to add facets to our projects, or change their direction, to accommodate their needs. Not only does this pique their interests, it also ensures the projects address the most pressing needs of Dell as an organization. To get this critical information, we simply had to ask. Stakeholders were often very happy to see us take an interest in their work. Unfortunately, we were at times guilty of failing to remember to ask this simple question as soon as we should have. Do not forget!

3.4.5 Find and Embrace Champions

We believe champions to be a key difference between a successful project and an unsuccessful one. A champion is someone who thoroughly embraces a project initiative, but not only that; a champion is also someone who has sufficient clout within an organization—through credibility, prestige, or a personal network—to help rally stakeholders behind it. Managing a project as transient outsiders, we found it difficult to define a vision others would think of as something other than naïve or a passing fad. Once a couple champions embraced our vision, however, they worked with us to refine it, extend it, and ultimately to motivate our core teams with it.

Identifying champions is not as difficult as it may seem. They often speak up in meetings, are spoken of highly by their colleagues, and may even seek you out if your ideas resonate with them. We were sure not to squander our opportunities with them, but rather ride their coattails and continually seek their advice. In fact, our insights in this section stemmed largely from the advice of these champions.

3.4.6 Keep Deliverables Simple

Any deliverables, such as agendas, documentation, spreadsheets, and even mathematical models, should be structured in as simple and straightforward a manner as possible. Taking additional time to strip out unnecessary content is highly worthwhile. Using terminology that is familiar to stakeholders, rather than that from a different industry or academia, is also important so stakeholders feel at ease and the learning curve is lowered. We are *not* advocating a “dumbing-down” of deliverables, but rather an investment in time to make these deliverables more readily digestible for time-pressed stakeholders. Our stakeholders were more willing to continue to discussions and use our tools when we made them intuitive.

Designing accessible, intuitive deliverables may be more of an art than a science. Either way, it takes practice and refinement. By gathering feedback from champions or some key stakeholders, we were able to iteratively improve our deliverables. Take our ARM model, for example. Our first version was drastically different from our final. Through subsequent iterations, we updated terminology to align with our stakeholders’ vocabulary (e.g. changing *COGS* to *standard cost*), deemphasized input fields they felt comfortable treating as constant (such as *inventory holding cost*), and making certain functional such as the *Generate recommendations* button more prominent and accessible without scrolling.

3.4.7 Post Content at Easily Accessible Locations

When dealing with global, cross-functional groups, we found it important to identify a virtual collaboration space where we could post core team materials. Ultimately, we think the technology platform or group managing it does not matter nearly as much as its accessibility. Burying content deep within a content management system (CMS) may pose problems with stakeholders who do not have the time to go looking for it. We advise keeping it simple by creating an easy-to-remember URL using a URL shortening service that directs stakeholders to the repository. In our case, we created the URL

<http://bit.ly/ArmModel> to direct stakeholders to the ARM model spreadsheet and User's Guide PDF, which GRL hosts on its Microsoft SharePoint CMS. Dell employees who type this URL in their web browsers see the appropriate content within GRL's site—regardless of whether they belong to GRL, GSP, or ARB—while individuals unauthorized to view the content see an error message. This is a great way to keep content easily accessible for all stakeholders while still secure.

3.4.8. Provide Training & Support

When developing tools for stakeholders to use, we recommend supplementing these tools with training and support. As we put the final touches on the ARM model, we assembled a User's Guide to address stakeholders' common questions, provide a step-by-step tutorial, and describe case studies; we show the pages from this guide in Figure 21. We also enlisted the help of a few employees to serve as a focus group; they helped us fine-tune the documentation and the usability of the model. Finally, we took the initiative in one-on-one meetings to provide a personalized walkthrough of the model tailored to individuals' interests. We found these efforts very important to the successful rollout of our initiatives. After on such training session, a GRL stakeholder said to us, "Before I told you I would use the model, but now I really mean it!"

Figure 21: Pages from the ARM Model User's Guide. We distributed the guide 23 stakeholders—from individual contributors to a vice president—covering ARB, GSP, and GRL in DAO, EMEA, and APJ.

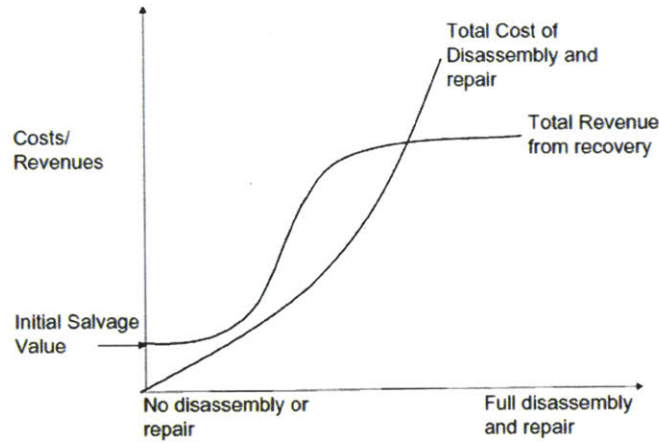


Chapter 4. Optimization System for Supply Chain Design

In collaboration with members of our working group at Dell, we devise an optimization system for the planning or tuning of a reverse supply chain. We call it the Asset Recovery Maximizer model (or ARM model for short). We consider such a system useful for many reasons, including the following:

1. The optimal solution is not obvious. Often, supply chain managers must choose among a robust menu of transformational/rework activities and recovery channels. They must determine not only which of these activities and channels to incorporate in their supply chain, but also the volume of product they should route through each of these channels. Lin shows the complexity behind one such deceptively simple question: what proportion of returns should be disassembled for parts? [29] We devise a system that addresses these and other questions at a macro level.
2. A robust, optimal solution strikes a balance between revenue and expenses. As Jayaraman illustrates in Figure 22, as costs in the reverse supply chain increase, so does total revenue, at least to a point. There is a sweet spot, past which there is no marginal benefit to additional expenditure. An optimization model can identify the fixed and variable expenses worthwhile to achieve maximal profit.
3. Time is an important factor. As Guide et al. demonstrate in their case study with HP computer returns, time is a critical factor in the reverse supply chain [34]. Particularly due to the rapid (and increasing) clockspeed of the IT industry, technology products experience swift price erosion as they grow obsolete. A reverse supply chain must balance cost-effectiveness with a swiftness to process returns quickly before their value erodes further. Guide et al. suggest reverse supply chains can manage this balance by finding the appropriate design between centralized and preponement [25]. Our model helps find this balance while simplifying the task of tracking product flows over multiple periods with depreciation, something difficult to compute by hand.
4. The future is uncertain. The demand uncertainty the forward supply chain faces is only one of the uncertainties in the reverse supply chain. The other key one is the volume of products that will actually be returned over time. Rework costs and resale prices are other uncertainties, but of course there are many others too. A model's sensitivity analysis can help supply chain designers get a handle on which factors are most critical in effecting overall profitability and key metrics such as net recovery. When these factors change, the designers know to reevaluate their decisions. Additionally, such sensitivity analysis can help these designers settle on a reverse supply chain design that is robust in the face of these uncertainties.

Figure 22: Dynamics between transformational expenses and recovery in the reverse supply chain, adapted from Jayaraman [35]



We begin by describing the model and its benefits at a high level, including the metrics it evaluates, the decisions it makes, and examples of situations where it can be applied. Then, we describe the core more precisely, in mathematical terms. We follow with the details of our implementation, covering input screens, the recommendation engine, and output screens. Next is a discussion of how we use the model to perform sensitivity analysis. We conclude by reporting on some of the feedback our model has received by its users at Dell.

4.1. Overview

The ARM model endeavors to answer these questions for a supply chain design and management team:

1. Of the transformational activities available to us—including various levels of repair, ODM exchange, and disassembly for parts, each with fixed enablement costs and variable costs—which ones should I enable?
2. Of the recovery channels available to us—including Outlet sales, auctions, and part usage, each with fixed enablement costs and variable costs—which ones should I enable?
3. Given anticipated product return volumes, what proportion of this returned inventory should undergo each type of transformation and be sold through these recovery channels?
4. How do our actions change in the face of shifting costs and market conditions?
5. How does this design measure up using the metrics of our business?

For members of our working group, encompassing ARB, GSP, and GRL organizations, answers to these questions are very important. We begin with the last question, by discussing the metrics these groups use for their decision making and monitoring of reverse supply chain health.

4.1.1. Key Metrics

The most important metrics used by these groups are net recovery, cost per box, cycle time, and inventory levels. We discuss each of them in turn.

4.1.1.1. Net Recovery

As previously discussed, net recovery is essentially an income-to-expense ratio. Revenue earned by a product or part is in the numerator, and its standard cost plus any transformational expenses in the denominator. Depending on the context in which the metric is used, the appropriate share of fixed costs may also be included in the denominator. Typically, when referring to an individual unit or a small population of returns, SG&A costs are excluded.

GRL and ARB groups use net recovery as a broad measure of health in the reverse supply chain. Especially when looking at trends or comparing with competitors, an increasing or higher net recovery metric indicates improved cost-effectiveness, increased demand for refurbished products and parts, or both; a decreasing number provides a warning to the opposite effect. A drawback to the metric is that it does not, on its own, help someone decode which is responsible for differences or changes over time.

4.1.1.2. Cost per Box

The cost-per-box metric appropriates the sum of per-unit transformational costs plus overhead costs—such as inventory and management—to the totality of units sold, thus providing a high-level measure of efficiency. The metric is quite intuitive, and useful at making comparisons over time. GRL sets cost-per-box improvements, and monitors the metric closely. Whenever cost per box increases unexpectedly, it triggers a deeper analysis into the cost components contributing to the rise. For better granularity, cost-per-box numbers can be reported separately by form factor or customer segment.

4.1.1.3. Cycle Time

With price erosion an ever-present threat, speed in the reverse supply chain is critical. GRL uses average and peak cycle time metrics to discern how long it takes inventory to move through a particular step in the process, or the process overall.

4.1.1.4. Inventory Levels

GSP, ARB, and GRL track inventory levels for their parts, and ARB and GRL also track inventory levels for their systems. There are three important places where inventory levels are closely watched:

1. WIP inventory: This inventory is in the *rework* stage of the “three Rs” process. Systems in this category are being transformed in some way, or in a queue to be transformed in some way.
2. AWP inventory: This inventory is a subset of WIP inventory for those systems awaiting one or more parts that are currently not available in the parts supermarket, with no suitable alternative parts available in the supermarket either. While Dell orders the parts

and waits for them to be delivered, the systems are placed on hold. They cannot make further progress through the reverse supply chain until the parts arrive.

3. Finish goods inventory: ARB, unlike DFS, places its CODP after the kitting step, at which point its systems are fully transformed into finished goods. It provides a staging area for these finished goods, where the goods sit until ordered through the outlet store or an ad-hoc deal ARB negotiates with a third party, typically when a high volume of a certain product resides in finished goods inventory. ARB has targets for the quantity of finished goods for each product family, based on sales rates, and occasionally engages in demand shaping via price changes on the outlet store to bring inventory levels in line with these targets.

4.1.2. Requirements

In conversations with our stakeholders, we identified some clear requirements for our model. We heard some explicitly, while we inferred others based on the way their respective roles and relative comfort with technology.

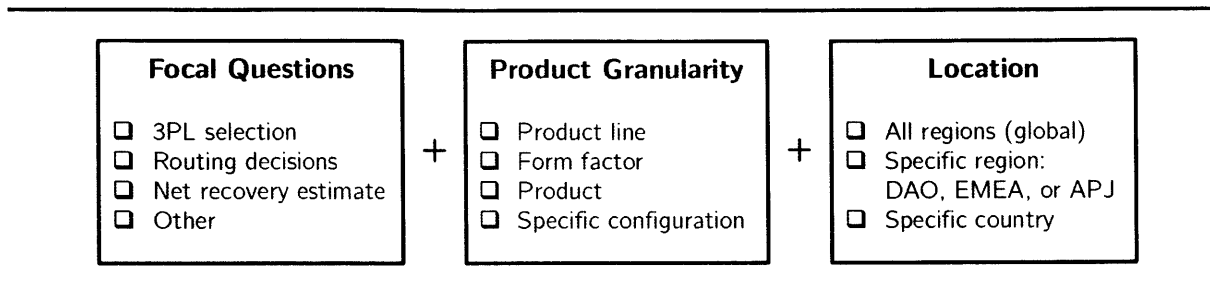
1. The model must be easy to use. Stakeholders do not have time to learn another software application, so the interface must be highly intuitive.
2. The model must be accessible from anywhere. Stakeholders work remotely, sometimes disconnected from the network. They should be able to use the model from remote locations without needing a high-bandwidth connection or even continuous network connectivity.
3. The model must run on standard-issue hardware. Stakeholders typically work on laptops with 2 GHz CPUs and 4 GB of memory; the model must run comfortably within these specifications.
4. The model must run quickly. Stakeholders do not have patience for long-running computations. Recommendations must appear nearly instantly.
5. The model cannot require any special software. Stakeholders may not have the administrative rights to install additional software, and, further, we do not have the budget to license software to such a large group.
6. Most importantly, the model must meet the needs of GRL, ARB, and GSP. We approached these groups with no preconceived notion of the purpose or structure of our solution. We listened to their challenges and ideas, and worked together in crafting the ARM model as a response to them.

4.1.3. Modeling Scenarios

The ARM model is sufficiently flexible to accommodate a variety of planning scenarios that GRL, ARB, and GSP may encounter. We advise users of the ARM model to define clearly each scenario of in-

terest before inputting anything into the ARM model, since the scenario dictates how they specify certain inputs and how they interpret the resulting recommendations. In Figure 23, we show the constituents of a scenario: one or more focal questions, a choice of product granularity, and a location.

Figure 23: Components of a modeling scenario



The *focal questions* component guides the user to relevant ARM model output: the go/no-go decisions section, financial impact, the network flow diagram, or the sensitivity analysis worksheet. We list the most common focal questions the ARM model is designed to address in Table 14.

Table 14: Typical focal questions of modeling scenarios

Category	Focal questions
Asset repair selections	<ul style="list-style-type: none"> Which types of repair work, ODM exchanges, etc. are cost-effective?
Asset recovery selections	<ul style="list-style-type: none"> Which recovery channels make financial sense?
Asset volumes	<ul style="list-style-type: none"> How many units will flow through each part of the reverse supply chain? How will these volumes change over time?
Net recovery calculations	<ul style="list-style-type: none"> What will be the net recovery rate for each rework activity and recovery channel? What will be the overall net recovery rate for the reverse supply chain?
Net income estimates	<ul style="list-style-type: none"> What is bottom-line financial impact to Dell? How sensitive is this impact to our forecast of returns, price erosion, or other factors?

The *product granularity* component helps the user input the correct information into product-specific fields like price and COGS. For coarser granularities, like form factor, we encourage users to use a weighted average price (WAP) and weighted average COGS. We define the most common granularities in Table 15.

Table 15: Typical product granularities of modeling scenarios, listed from broadest to narrowest

Granularity	Example
Product line	Alienware
Form factor	SFF devices (combination of Mini 3i, Aero, Streak, and Venue Pro)
Product	Streak (both 16GB and 32GB configurations)
Specific configuration	Streak (16GB configuration only)

Finally, the *location* component puts the user in the appropriate mindset to specify the rework activities, recovery channels, and cost and price information applicable to the focal country or region. The model does not accommodate multiple regions simultaneously, so if an area under consideration cannot be treated holistically, perhaps because certain activities or channels are not available in certain areas, or logistics costs differ among the areas, the ARM model must consider the areas separately. Due to the heterogeneity among countries in the APJ region, specifically, we see a need to focus on specific countries or smaller clusters of countries sharing similar labor rates and logistics costs. Other times, as with EMEA, we were able to conduct a successful case study considering the entire region as a homogenous unit.

4.1.4. Benefits

To our working group stakeholders, we posit the following benefits of using the ARM model:

- ARM saves you time. For any scenario you specify, ARM is able to calculate optimal recommendations within 1-2 seconds. You can easily revise your assumptions, or add in a new vendor, and see the impact just as quickly.
- ARM is your personal accounting assistant. ARM calculates net recovery to accounting's specification, and also knows how to distribute set up costs and rework costs into recovery channel cost-per-box calculations. These calculations may be tedious or difficult to do by hand.
- ARM helps you track changes over time. Over time, price erosion (how the price of a product declines over time as its technology becomes dated), cost erosion (the tendency for material costs to decline over time), and inventory holding costs have a big impact on the bottom line. ARM tracks the flow of product over time, quarter-by-quarter, keeping all of these factors in mind – so you do not have to.
- ARM keeps you organized in the face of uncertainty. The future is uncertain. What will the return rate for a product be in Q2? What will the true price of a level 2 repair be? The ARM model helps you structure your thoughts you can not only lay out all your assumptions on the table, but also give you recommendations that deal with a range of possibilities.
- ARM makes collaboration easier. The ARM model provides a standardized decision-making process that works across all regions, so you can share your analyses with other regions and more easily interpret their analyses as well.

4.2 Model Formulation

In this section, we describe the mathematical formulation behind the ARM model's recommendation engine. Our formulation is a MILP, meaning 1) we express the objective function and all constraints

linearly with respect to the decision variables, and 2) we require some of the decision variables—specifically our binary decision variables—to be integers, namely 1 or 0. We use a network flow problem as a prototype for the formulation since it accurately describes the type of environment we wish to model: a network, namely the reverse supply chain, with returns flowing through the network.

4.2.1. Network Flow Problem

A network flow problem has these features:

- A set of nodes, each of which may be associated with a supply and/or demand of some number of units.
- A directed set of arcs, or pathways, from source nodes to destination nodes. Each arc may have a capacity constraint and a per-unit transfer cost.
- Decision variables that describe how many units flow from a node with supply to a node with demand from each directed arc.

Table 16 shows how we relate these general features to the specific features of a reverse supply chain. In our mapping, we model the arrival of product returns into the network by adding units to the appropriate supply nodes based on the condition of the return. For example, when a customer returns a DOA product, the number of units at the DOA node increases by one. While, in reality, Dell does not necessarily organize returns by their respective conditions—with a bin for DOAs and a bin for BERs, etc.—the mapping provides an elegant way for us to track inventory through the reverse supply chain as a network flow problem.

We model the departure of product returns (and parts) out of the network by moving units to the appropriate demand nodes. We add constraints to ensure all units eventually move to demand nodes, so that everything in the reverse supply chain ends up sold. This does not mean all units move directly from supply nodes to demand nodes, however. Some units may move from one supply node to another supply node. For example, an OEM exchange is modeled as the movement of a unit from a *DOA* node to an *unopened/new* node in the event the OEM exchanges the defective unit for a new one, and to a *refurbished* node when the OEM exchanges the defective unit for a refurbished one. No units flow out from a demand node once they have entered it, however, since a demand node represents the sale or usage of that unit.

Table 16: Network flow formulation features in the ARM model

Network flow formulation feature	Analogy to the reverse supply chain in the ARM model	Examples
Supply nodes	Product inventory by return condition	<ul style="list-style-type: none"> • BER • DOA • L1 (i.e. in need of L1 repair to refurbish) • L3 (i.e. in need of L2 repair to refurbish) • L3 (i.e. in need of L3 repair to refurbish) • Refurbished • Unopened/new

Network flow formulation feature	Analogy to the reverse supply chain in the ARM model	Examples
Demand nodes	Recovery channels	<ul style="list-style-type: none"> • Auction sale • Outlet sale • Part harvesting for sale or internal use • Revenue share with third-party outlet • Seed stock replenishment • Warranty stock replenishment
Arcs	Transformational activities	<ul style="list-style-type: none"> • L1, L2, L3, etc. repair, transforming defective products into refurbished products • ODM exchange, transforming DOA products into new (or refurbished products) • Teardown, transforming whole products into parts

The examples we cite along with each network flow feature in Table 16 are, just that, examples. The ARM model implementation accommodates *arbitrary* return conditions, recovery channels, and transformational activities that link the two. The ARM model dynamically forms the arcs in the network based on user specifications with each run of the model. This allows the model to cover a broad range of products and reverse supply chain capabilities. For example, the model comfortably accommodates the ODM exchange option of SFF devices and a varying number of quality levels.

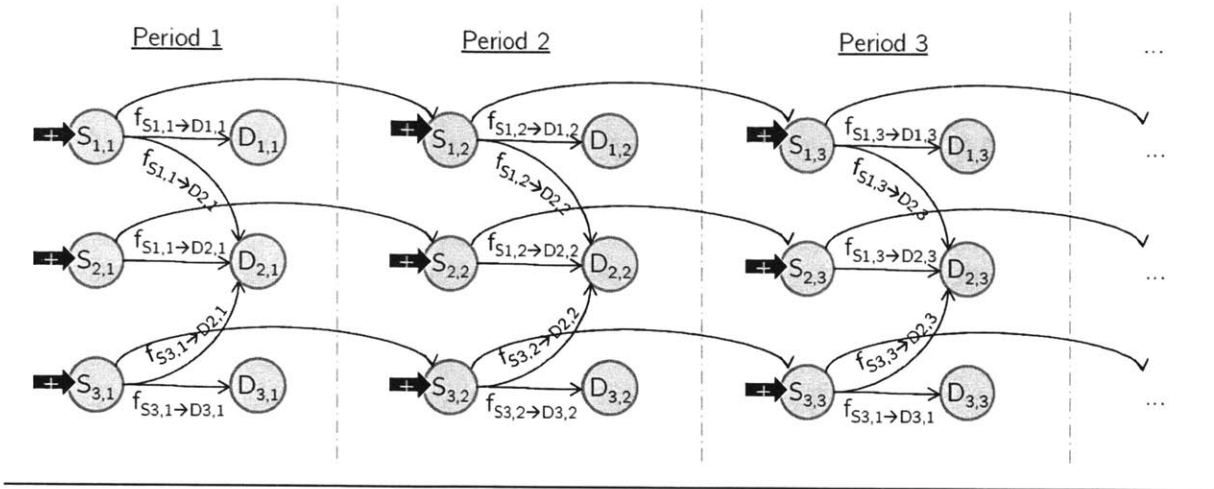
The ARM model uses whole numbers when expressing return volumes. If a user specifies that 20% of a sales volume of 14 units will be returned, the ARM model does not express the number of returns as $20\% \times 14$, or 2.8; instead it rounds to 3. For typical modeled return volumes, which are in the thousands of units or greater, rounding has no effect on the outcome. It does offer an important optimization, however. Network flow problems with supply units expressed as whole numbers, and capacity constraints expressed the same way, take advantage of an integrality property

In addition to supply nodes, demand nodes, and arcs, we include several extensions to the typical network flow-style formulation: multi-period support, product quality restrictions, holding cost accounting, and arc and node enablement. We describe these extensions in the following subsections.

4.2.1.1. Multi-Period Support

Our literature review reveals the importance of reverse supply chain design to the sensitivity of time. Over time, prices erode, as do their material costs, while inventory holding costs and cycle time within the reverse supply chain prove to be factors affecting net recovery. Return rates vary over time. Demand through various recovery channels also varies. Instead of modeling a one-time network flow problem, we need to provide for the movement of product over time.

Figure 24: Network flow problem with a time component



As we show in Figure 24, we extend the network to support multiple periods by creating multiple copies of the network, one for each period. Each node and arc of one period has a copy in each other period, with each copy referenceable by a time-based index. Given n periods and supply node S_1 , for example, we define n copies of S_1 , identified as $S_{1,1}$; $S_{1,2}$; ...; $S_{1,n}$. The same is true for arcs. If an arc extends from S_1 to D_1 , we define n copies of it, denoted as $f_{S_{1,1} \rightarrow D_{1,1}}$; $f_{S_{1,2} \rightarrow D_{1,2}}$; ...; $f_{S_{1,n} \rightarrow D_{1,n}}$.

To ensure continuity between periods, we draw additional arcs from supply nodes each period to their respective chronological successors, such as an arc from $S_{1,1}$ to $S_{1,2}$. Additionally, we force all material in a period to a demand node or successor supply node by adding constraints to set the final inventory level to 0 for all supply nodes.

Conveniently, this approach towards modeling multiple periods makes a couple time-based constructs straightforward:

1. We can define time-based capacity constraints, associated with transformations or demand saturation in a reverse supply chain recovery channel, simply by adding constraints to the relevant arcs. If, for example, we want to limit material flowing from S_1 to D_1 at x units per period, we add a capacity constraint of x to each of the arcs.
2. We can define time-based product returns by setting the starting inventory level of the appropriate supply nodes. If returns of type 1 start at 100 in period 1 and grow to 200 in period 2, for example, we set the starting inventory level of $S_{1,1}$ to 100 and of $S_{1,2}$ to 200.

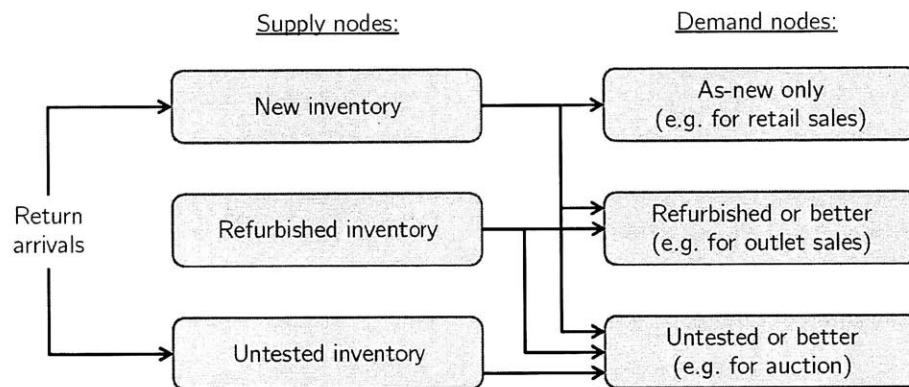
4.2.1.2. Product Quality Restrictions

While our formulation considers only a single type of product, we do accommodate multiple return conditions, such as unopened/new, DOA, and the others we list in Table 16. We must keep track of these differentiated products separate; the recovery channels are very particular about the quality they permit. Consider the Dell Outlet: a product must meet a certain quality standard to be listed on the

outlet. Dell typically sells refurbished products through its outlet, but it sometimes offers unopened/new inventory as well. In fact, when browsing through items in the outlet, Dell indicates the quality level of the item and prices it accordingly. At the same time, it does not list items of other quality levels, such as DOAs, untested inventory, or inventory in a state of disrepair.

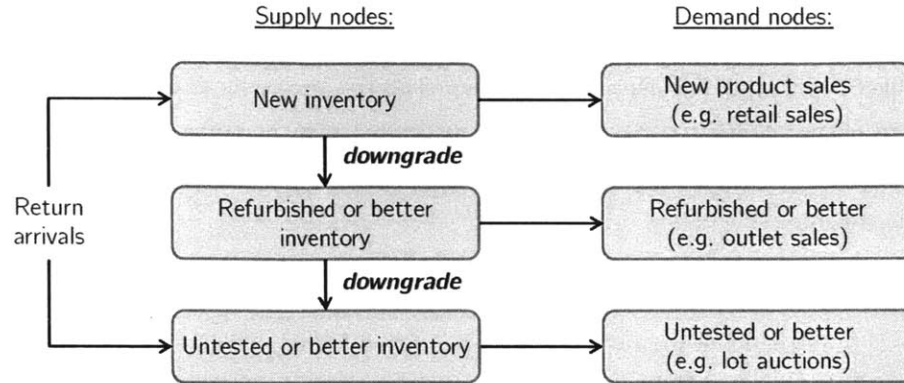
We offer a simple way to account for these quality restrictions: arcs. With supply nodes representing inventory at the supported quality levels, we draw arcs to the demand nodes only when the demand node permits products at that quality level. Figure 25 shows a simplified network graph for illustrative purposes. Here we permit new inventory to flow to any of the three demand nodes, but are less permissive with refurbished inventory, and less permissive still with untested inventory.

Figure 25: Flow network with product quality constraints



In cases where a recovery channels offers the same revenue opportunity regardless of product quality, we offer an optimization to the network. We can reduce the total number of arcs by defining a “downgrade” process where we draw arcs from supply nodes to other supply nodes representing lower-quality inventory. Then, we are able to limit the supply-to-demand arcs to those with starting points on the supply nodes representing the minimum quality level permitted by the respective endpoints. In Figure 26, we present the alternative to the flow network depicted in Figure 25. With this particular network, the optimization reduces the total number of arcs from six to five. In more complex networks, the reduction can be much greater.

Figure 26: Optimized flow network using downgrade process



The downgrade process works well because of our model's objective, profit maximization. Demand nodes restricted to higher quality levels generally offer better prices than their more permissive peers. Consequently, in the optimal solution, high-quality inventory is typically routed to these more lucrative demand nodes until saturated; then, and only after that point, any remaining such inventory is downgraded, treated as lower quality, and routed to the less lucrative demand nodes along with the other lower-quality inventory.

We take advantage of this optimization in our implementation, though it does require us to impose a restriction: the demand node cannot vary the price of the good based on quality. In cases where quality has an effect on price, we recommend creating additional demand nodes targeted to different quality levels. Using this optimization conveys another benefit: since it implies only one arc enters each demand node, the capacity constraints we set on these arcs express the total capacity of the channel rather than simply the capacity of the channel for a specific product type. This is ideal, because per our previous assumption, the channel treats all products identically regardless of quality and has an overall capacity.

4.2.1.3. Holding Cost Accounting

In the traditional network flow problem, costs are associated with the arcs only. As products move from one node to another, the objective function reflects an accrued variable cost. In our case, we also account for inventory holding costs, which are not reflected by the movement of product through the network, but rather the lack thereof. We extend the network by charging for inventory that lingers in the supply nodes from one period to the next using a user-supplied inventory holding cost based on a percentage of COGS over the course of the period.

4.2.1.4. Arc and Node Enablement

Our final extensions to the flow network are the binary decision variables we attach to the transformational arcs and demand nodes. These binary variables represent go/no-go decisions, similar in spirit

to the decision variables we discuss in our literature review related to facility placement. In our case, the decisions are whether to engage in particular transformations and whether to pursue certain demand nodes (i.e. recovery channels). We structure the formulation such that 1) these binary variables are set to 1—or “enabled”—when product flows through the respective arc or flows to the respective demand node, and 2) these binary decision variables may be associated with one-time charges we can consider fixed costs or set-up costs. These costs are incurred only when the transformational activity or the recovery channel is enabled.

4.2.2. Model Parameters

In this and the following subsections we provide a formal definition of our model. We begin with model parameters, listed in Table 17. Our model treats these parameters as constants, though a user running our model has the opportunity to review and change them before running the branch-and-bound (B&B) algorithm to find an optimal solution. The parameters cover the following dimensions:

- Price and costs associated with the product, and associated price and cost erosion. For price erosion, we supply a default of a 1% decline in price per week, consistent with Blackburn et al.’s research into the price erosion of electronics [26]. For cost erosion, we supply a default of a 0.25% decline in cost per week, based on a colleague’s research findings at Dell [50].
- Revenue earned through the various recovery channels, expressed in any combination of a percentage of COGS, percentage of sales price, or a fixed amount.
- Inventory holding costs, expressed as a percentage of COGS per period. In our specific implementation, we use quarterly periods, though the formulation is generic enough to use with any desired temporal granularity.
- Fixed and variable costs associated with transformational activities and recovery channel. The formulation assumes the fixed costs associated with any transformation activity or recovery channel is not incurred by the firm unless the node is enabled.

Table 17: Model parameters

Parameter name	Description
$w_{i,t}$	volume of product returned to node i in period t
v_i	maximum allowable volume of product that may flow to node i in any given period
f_i	fixed revenue earned when a unit flows to node i
r_i	relative revenue earned when a unit flows to node i , expressed as a percentage of the current price
q_i	relative revenue earned when a unit flows to node i , expressed as a percentage of COGS
p_t	relative price of a unit at period t
c_t	standard cost (COGS) of a unit at period t
h_i	1 if an inventory holding cost is incurred at node i ; 0 otherwise
s_i	set-up cost to enable node i

Parameter name	Description
d_i	variable cost of sending a unit to node i
z_i	quarterly inventory holding cost of capital, expressed as a percentage of COGS

The parameters are flexible enough for a user to focus at any level of granularity. The COGS variables, for example, may refer to an entire form factor (broad), product line, product, or specific configuration (narrow). In our case studies, we take advantage of this. Our SFF case study in Chapter 5 is form factor-based, while our Alienware repair case study in Chapter 6 is product line-based. While at Dell, we also applied the model successfully to product and specific configuration scopes, such as to the Dell Streak smartphone (product) and the black Dell Streak 16GB smartphone (specific configuration).

4.2.3. Decision Variables

The formulation's decision variables are used to make two types of decisions: first, how to route returns through the supply chain—which determines the appropriate mix of transformational activity and recovery channel volumes, and, second, which rework activities and recovery channels to enable.

Table 18: Decision variables

Variable name	Description
$X_{i,j,t}$	volume of product flowing from node i to j in period t
Y_i	1 if rework activity or recovery channel i is enabled; 0 otherwise

4.2.4. Auxiliary Variables

The auxiliary variables we define here are to simplify how we express flow constraints mathematically. We do not have a need for them in our actual implementation.

Table 19: Auxiliary variables

Variable name	Description
$S_{i,t}$	remaining stock of product at node i in period t , after considering any inflows to i and outflows from i within the period

4.2.5. Objective Function

In our objective function, we attempt to maximize profit by taking into account revenue, variable costs, and fixed costs. Thus, our model finds the optimal balance between rework activities and recovery channels that maximize revenue while at the same time minimizing cost. We do not maximize *net recovery*, ARB's preferred metric and one we discuss in detail earlier, for several reasons:

1. We cannot express net recovery as a linear objective function.

2. As we discovered during our SFF case study and will discuss further in Chapter 5, net recovery is not a suitable proxy for profitability. The two metrics may, in fact, move in opposite directions.
3. A net recovery-based objective function may have multiple global optima associated with supply chain designs having varying degrees of efficiency. A profitability-based objective function, on the other hand, is guaranteed to have an unambiguous, unique global optimal.

We also do not minimize *cost*, unlike most of the models we review in Chapter 2 do. In our view, cost does not provide a comprehensive or mature view of the reverse supply chain, which Stock et al. compel us to see as a profit center [9].

We express our profit-maximizing objective function as follows:

$\text{Maximize } \sum_i \sum_j \sum_t X_{i,j,t} (r_j p_t + q_j c_t + f_j)$	<i>revenue</i> (4.2.5.1.)
$- \sum_i \sum_j \sum_t (X_{i,j} d_i + S_{j,t} c_t h_j z)$	<i>variable costs</i> (4.2.5.2.)
$- \sum_i Y_i s_i$	<i>fixed costs</i> (4.2.5.3.)

4.2.5.1. Revenue

The revenue component of the profit function captures all types of revenue—price-based, COGS-based, and fixed—earned as products are routed to the various recovery channel nodes, which models sales through these channels.

4.2.5.2. Variable Costs

The variable costs component accounts for per-unit expenses incurred in transformational activities, such as repair costs, and recovery channels, such as any shipping and handling charges or other 3PL fees paid by Dell.

4.2.5.3. Fixed Costs

The fixed costs component of the profit function accounts for one-time set-up costs associated with leveraging a particular transformational node or recovery channel node. If and only if zero units of product flow through the node across all periods, the cost is not incurred.

4.2.6. Constraints

The constraints in our formulation maintain the integrity of the flow network, ensure each return ends up sold through a recovery channel, and accommodate any limits a user might place on the amount of the product flowing through any node at any point in time. In total, we define seven types of constraints, which we describe in detail following their mathematical expression:

$R_i \in \{ 0, 1 \}$	$\forall i$	<i>binary</i> (4.2.6.1.)
$X_{i,j,t} \geq 0, S_{i,t} \geq 0$	$\forall i, j, t$	<i>non-negativity</i> (4.2.6.2.)
$S_{i,t} = S_{i,t-1} + \sum_j (X_{i,j,t} - X_{j,i,t}) + w_{i,t}$	$\forall i, t$	<i>network flow</i> (4.2.6.3.)
$\sum_j X_{j,i} \leq v_i$	$\forall i, t$	<i>capacity</i> (4.2.6.4.)
$S_{i,t=\text{final period}} = 0$	$\forall i$	<i>depletion</i> (4.2.6.5.)
$\sum_j \sum_t X_{i,j} \leq MY_i$	$\forall i;$	<i>big-M</i> (4.2.6.6.)

4.2.6.1. Binary Constraints

The binary constraints ensure that the enablement decision variables Y_i each take on a value of either 0 (disabled) or 1 (enabled). This constraint is solely responsible for changing the character of our formulation from a simple linear program (LP) to a mixed-integer linear program (MILP). The other decision variables are not constrained to be integers, as that quality arises naturally from the integrality property of flow networks.

4.2.6.2. Non-negativity Constraints

The non-negativity constraints ensure that the volume of flow leaving a node is always non-negative, and the ending stock at any node is always non-negative.

4.2.6.3. Network Flow Constraints

The network flow constraints maintain the integrity of the flow network, ensuring, essentially, conservation of mass throughout the network. It states that the volume at a node in a given period is the same as its volume in the prior period, less any outflows to other nodes, plus any inflows from other nodes, plus any relevant returns received in the period.

4.2.6.4. Capacity Constraints

The capacity constraints limit the amount of product flowing between any two nodes in any period. There are two types of arcs to consider:

- Arcs from a source node to another source node: These arcs imply a change in quality level, either through a transformational activity, such as repair, or a downgrade. For arcs representing transformational activities, the capacity constraints signify a limit to the number of products it can transform in a given time period due to limited inventory, machines, personnel hours; company policy; or perhaps contract terms. For arcs representing downgrades, our implementation imposes no capacity constraints, as downgrades our virtual constructs that require no actual physical activity and implies no additional time or cost.

- Arcs from a source node to a demand node: These arcs imply a sale of inventory through a recovery channel. Here a capacity constraint signifies a limit to the amount of inventory that may be sold through the channel in a given period, due to demand saturation, contract terms, or perhaps strategic reasons.

4.2.6.5. Depletion Constraints

The depletion constraints ensure that that no inventory remains in supply nodes at the end of the last period. In other words, we ensure all inventory is eventually sent to a recovery channel. This mirrors reality. At no point would Dell indefinitely accumulate inventory, even if it had no value; it would find some recovery channel, even if that happened to be a scrap or recycling channel offering zero revenue or even a cost.

These constraints may render any solution to a particular supply chain network infeasible. If product is returned in a state that no recovery channel will accept, then there is no transformation by which that product can leave a supply node, violating the constraint. Even if such transformation exist, if the volume of product is greater than the capacity of the transformation to move the product to the demand node by the last time period, the result will still be an infeasible solution. In our implementation, we inform the user when these situations arise. They have two ways to remedy the situation: increase capacity, or introduce additional recovery channels accepting of inventory at the quality level of what remains.

4.2.6.6. Big- M Constraints

The “Big M method” is a linear programming technique we employ to ensure the Y_i binary decision variables assume the value of one (i.e. “enabled”) when any volume of product flows through node i in any period, and are permitted to be zero (i.e. “disabled”) otherwise. In our implementation, we choose a value of M sufficiently large such that it exceeds the sum of all product flows across all arcs in all time periods.

4.2.7. Computational Complexity

In Chapter 2, we identify many of the optimization models reviewed in Chapter 2 as belonging to a class of computationally complex problems called NP-hard. The time required to find optimal solutions to these network design problems grow exponentially as the number of nodes in the network grows linearly [51]. We speculate that our formulation—which makes similar decisions by determining which transformational activities and recovery channels to enable—is one of the few in the class of MILP problems to be solvable in polynomial time, at least in the average case.

Our formulation is expressed simply enough, with a limited number of nodes and arcs, and the right-hand-side of all constraints are integers. Thus, it tends to be solvable quickly, in the order of a few seconds or less on typical computer hardware, such as a Dell employee’s company laptop. This affords us

the ability to run with many permutations of inputs without a significant time penalty, a cornerstone of our approach towards sensitivity analysis.

4.3 Implementation Overview

We implement the ARM model as a Microsoft Excel spreadsheet with 10 worksheets. We make four worksheets visible by default. The rest appear when users wish to supply additional input or wish make changes the underlying recommendation engine. Table 20 summarizes these worksheets, while the subsequent sections of this chapter describe them in detail; they are organized by category: user inputs (4.4), recommendation engine (4.5), and outputs (4.6).

In addition to an implementation of the MILP, the worksheets also provide complementary calculations and tools, including:

1. Financial calculations,
2. Cost-per-box calculations,
3. Benefit-per box calculations,
4. Implementation recommendations,
5. Network flow diagram visualizations, and
6. Sensitivity analysis

Within the worksheets, we embed macros written in the Visual Basic for Application Script (VBA) language to automate some tasks, including:

- Running the mixed-integer solver using a B&B algorithm to find optimal solutions,
- Generating the graphs that visualize the dynamically-generated reverse supply chain network,
- Calculating benefit-per-box, and
- Performing multi-scenario sensitivity analysis.

We catalog these macros and present a partial listing of the corresponding VBA scripts in Appendix B.

Table 20: Overview of the 10 worksheets constituting the ARM model

Category	Worksheet name	Purpose	Visibility
User input (4.4)	Input screen (4.4.1)	Provides the areas to specify inputs pertaining to the “three Rs” of the reverse supply chain—returns, rework, and recovery—for the scenario under consideration.	Always visible
	Production schedules (4.4.2)	Provides a mechanism for users to review, edit, and define new production schedules. A <i>product schedule</i> indicates the timing of new product sales in the market, which are then used as the basis for estimating return timing and volumes.	Appears by clicking an “edit” button on input sheet
	Depreciation profiles (4.4.3)	Provides a mechanism for users to review, edit, and define new depreciation profiles. A <i>depreciation profile</i> indicates the rate of price or cost erosion over time. We pre-define profiles to reflect constant erosion, accelerating erosion, and diminishing erosion at various rates, as well as a profile to express no price erosion at all.	Appears by clicking an “edit” button on input sheet
	Recovery channel worksheet (4.4.4)	Provides a mechanism for users to itemize costs and revenue sources related to a specific recovery channel. The worksheet includes a variety of fields unique to specific types of products, like carrier bounties for smartphones and rebate coupons for listings on Dell.com or the Dell Outlet.	Appears by clicking an “edit” button on input sheet
Recommendation engine (4.5)	Nodes (4.5.1)	Defines the nodes in the reverse supply chain network. There is a dynamic aspect to the network definition based on inputs specified on the Inputs worksheet.	Not visible by default; intended for advanced users
	Formulation (4.5.2)	Implements the MILP formulation defined in section 4.2 using cells to express the objective function, decision variables, and constraints, as well as the Frontline Systems’ Standard Excel Solver add-in to run its B&B mixed-integer optimization algorithm.	Not visible by default; intended for advanced users
Outputs (4.6)	Recommendations (4.6.1)	Issues the key reverse supply chain design recommendations, including “go” or “no-go” decisions for each rework option and recovery channel, as well as product routing recommendations with volume estimates, a financial summary, benefit-per-box calculations, and net recovery estimations.	Visible by default; shown when recommendations are generated
	Cost attributions (4.6.2)	Distributes fixed costs among recovery channels for use with the net recovery calculations.	Not visible by default; intended for advanced users
	Network flow (4.6.3)	Illustrates the supply chain design and routing decisions visually.	Always visible
	Sensitivity analysis (4.6.4)	This sheet is for understanding how recommendations might change in the face of uncertainty or variability. It lets you specify multiple inputs in a single place, and view the corresponding recommendations in a PivotTable and PivotChart. For example, it can show you recommendations across a range of possible repair costs or return rates.	Always visible

4.3.1 Typical Process

To use the ARM model, we instruct users of the ARM model to follow these steps:

1. Define or verify the modeling scenario to evaluate, as described in Section 4.1.3. Think about the specific question(s) you are trying to answer before diving into the model.
2. Download the ARM spreadsheet from its location in Dell's CRM system, or, if collaborating with others, save the model email attachment sent to you.
3. Open the model in Microsoft Excel. To record changes, enable Excel's revision tracking feature.
4. Fill out the input screen based on the scenario defined in step 1.
5. Click the "Calculate Recommendations" button to tell the ARM model to generate the reverse supply chain design recommendations and corresponding financial analysis
6. View the recommendations and network flow diagram.
7. If desired, conduct more advanced analysis, such as:
 - a. Computing the benefit-per-box for transformational activities and recovery channels, and/or
 - b. Performing a multi-scenario sensitivity analysis

4.3.2 Limitations

Early on, we determined our stakeholders wanted the capabilities of the ARM model without the need for downloading or installing any special software. Given this requirement, we chose to implement our model using Microsoft Excel with the Frontline Systems' Standard Excel Solver add-in. Both Microsoft Excel and the Solver add-in, which is bundled with Excel, are pre-installed on our users' machines. Unfortunately, the bundled version of the Solver add-in, unlike its premium counterpart, limits the number of decision variables and constraints we may leverage in our formulation. Consequently, we must make some tradeoffs as we scope the size of the network to meet the users' needs while adhering to these limitations. A nice side effect of this effort, however, is a concise formulation to which optimal solutions may be found quickly.

In Table 21, we summarize the number of decision variables and constraints we use in our formulation. Solver's exact limitations are a bit ambiguous, as it treats certain constraints differently from others, but we found that we cannot add any more constraints to our model—beyond the 207 already defined—with our non-premium version of Solver.

Table 21: Summary of variables and constraints used in the formulation

Formulation component	Total number of items	Breakdown
Decision variables	106 variables	<ul style="list-style-type: none"> • 16 binary variables for enabled/disabled decisions • 90 network flow decision variables
Constraints	207 constraints	<ul style="list-style-type: none"> • 16 binary constraints • 90 non-negativity constraints • 32 network flow constraints • 45 capacity constraints • 8 depletion constraints • 16 big-M constraints

Our limited space for decision variables and constraints translates into the following restrictions for our users:

- Maximum of 5 periods of recommendations. The ARM model's spreadsheet is formatted to display up to 8 quarters for product flow, and, indeed, the formulation itself may work with an arbitrary number of quarters, but due to Solver limitations we are only about to project product flows for 5 periods, or quarters. All returns must arrive by period 5 for a feasible solution to be found.
- Capacity constraints are not supported for the first three repair options. As shown in Figure 27, we do not permit constraints for these options simply because we not have room for additional constraints in our formulation. There is a workaround for a specific case, though; users who wish to disable one of these rework options, instead of specifying a capacity of zero, may express it by specifying a variable cost-per-box or fixed set-up cost that is exorbitantly high, such as \$999,999. This will dissuade the ARM model from performing the rework activity, which effectively disables it.

Figure 27: Location of capacity constraint limitations

Rework	Inventory		Costs		Capacity
	Rework method	Input type	Output type	Cost per box	Set-up cost
	Level 1 repair	L1	Refurbished	\$25.00	\$1,000.00
	Level 2 repair	L2	Refurbished	\$50.00	\$2,000.00
	Level 3 repair	L3	Refurbished	\$75.00	\$3,000.00
	ODM Exchange	DOA	New	\$5.00	\$0.00
	Teardown	BER	Parts	\$25.00	\$1,000.00

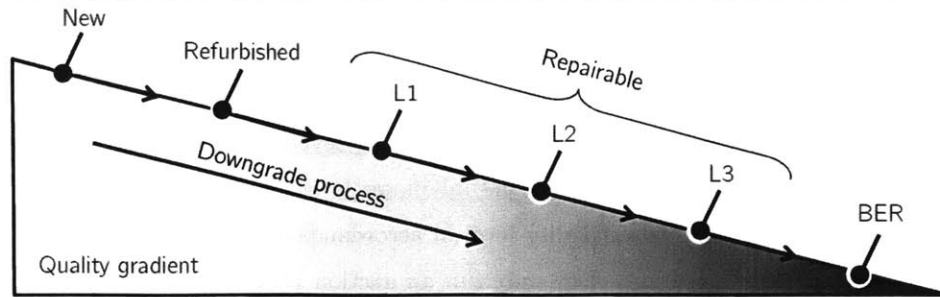
We believe we have found a good balance of functionality given the restrictions at hand. Of course, users who upgrade to the premium version of the Solver add-in may overcome these limitations. For a small group of users, the costs associated with this or other powerful optimization software may range from a thousand dollars to tens of thousands of dollars.

4.3.3 Assumptions

Naturally, we are unable to capture all of the complexities of a real-world reverse supply chain in our model, plus we contend with the additional constraints of the Solver add-in. Out of necessity, then, we make a few simplifying assumptions, including the following:

1. Recovery channels require a minimum quality level, and they treat products above this minimum level equivalently to the minimum level. Our model implicitly “downgrades” inventory as this higher-quality level in accordance with the quality gradient we show in Figure 28. For example, Dell may run an auction process to sell its BER inventory at a low cost. If it has excess new inventory, it may also sell this inventory through an auction. While, in reality, it will likely be able to command a higher price for a lot containing some functional equipment, our model does not consider this. We make this simplification to minimize the number of arcs in our formulation for efficiency.
2. Rework activities happen in the same period (quarter) products are routed to the rework stations, up to the maximum capacity of the respective rework stations. For simplicity, the model assumes all parts for repair work are readily available, so systems will not need to idle in awaiting parts (AWP) inventory. Inventory beyond this capacity incurs a quarterly inventory holding cost.
3. Resale activities happen in the same period (quarter) products are routed to the recovery channels, up to the maximum capacity of the respective rework stations. Inventory beyond this capacity incurs a quarterly inventory holding cost.
4. Returned products are valued at standard cost, even if the market price is below standard cost. This is consistent with Dell’s accounting policies. We offer the alternative lower-cost-or-market (LCM) valuation in the model’s financial summary section, but the objective function is not based on the LCM valuation. As standard cost may change by quarter due to cost erosion, ARM uses the appropriate standard cost for a product considering the period in which a customer returns it, not the period in which Dell manufactures it; which, again, is consistent with Dell accounting policies.
5. All units must be routed to a recovery channel by the final period. We do not permit any returns to linger in the reverse supply chain—such as AWP WIP—past this period.
6. Price erosion and cost erosion are based on percentage declines, not fixed-unit declines. The ARM model allows users to customize how this percentage may change period-by-period, such as 5% in the first period and 4% in the next, but it does not allow users to specify fixed values such as \$50/period.

Figure 28: Quality gradient depicting the implicit downgrade process in the ARM model



4.4. User Inputs

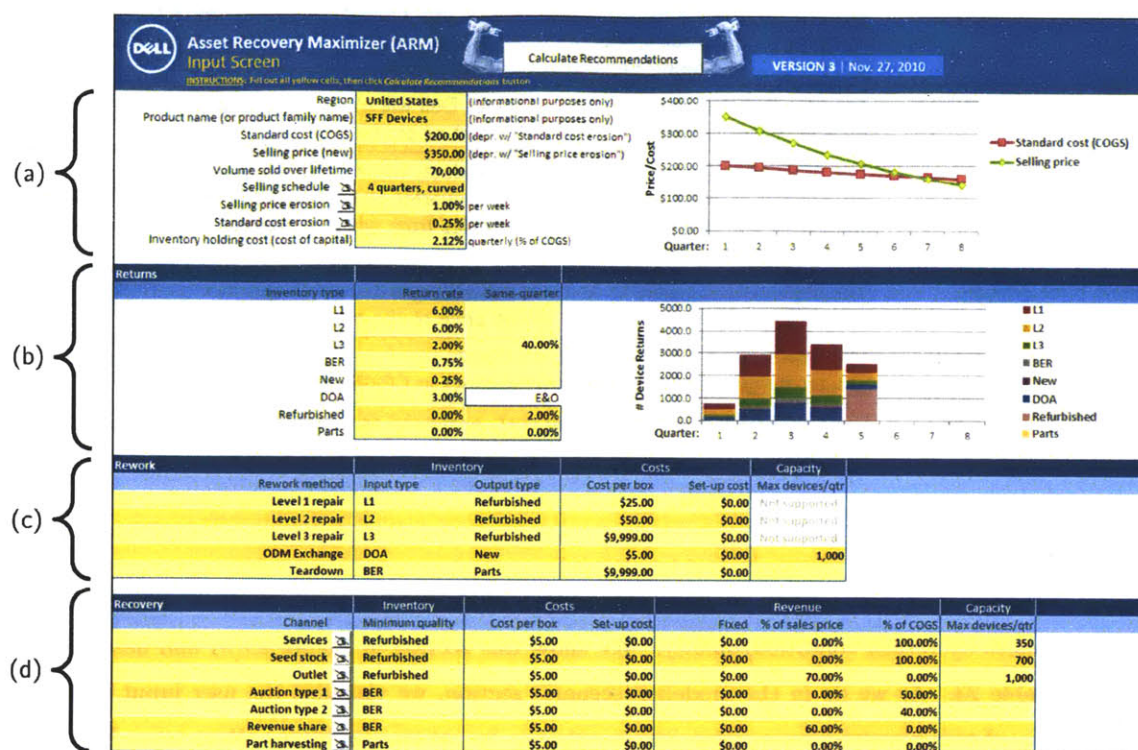
In this section, we describe the main input worksheet, naturally named the *Input Screen*, and the auxiliary input worksheets to edit production schedules, depreciation profiles, and recovery channels.

4.4.1. Input Screen Worksheet

We show the input screen worksheet in Figure 29. We identify all input fields with a yellow background. Some of our users, struck by its simplicity, ask for the underlying data sources to which the ARM model links. In truth, there are no such data links or system dependencies. In this sense, the model is straightforward. Some of the fields, however, may ask a user for information he or she may not know—like the standard cost of a product, for example. In these cases, the user must query the appropriate employee or database for the answer; the model itself does not provide lookup functionality.

We design the input screen to use terminology familiar to Dell employees. This permeates to the overall structure of the screen as well: after a few questions about the overall modeling scenario, we organize inputs into the “three Rs”: returns, rework, and recovery. We discuss each of these sections in turn.

Figure 29: Input screen worksheet, consisting of (a) general scenario and product information followed by the “three Rs”: (b) returns, (c) rework, and (d) recovery



4.4.1.1. Modeling Scenario Section

The first section of the inputs screen is where users describe the modeling scenario. Some of the fields are purely informational to walk through the exercise of clearly identifying their focal questions, product granularity, and location of interest, as we depict in Figure 23. See Table 22 for a description of the input fields in this section. Also note the graph in Figure 29(a): this allows the user to see how their inputs affect the new sales price and standard cost over time. As they specify these values and choose price and cost erosion percentages, the graph allows them to inspect and verify their choices.

Table 22: Explanation of input fields in the modeling scenario section

Input field name	Explanation
Region	The location under consideration, which may be very broad (all regions) a specific region (e.g. EMEA), or even a specific country (e.g. U.S.). For informational purposes only; does not influence ARM model recommendations.
Product name	The name of the product line, form factor, product, or specific configuration to evaluate. For informational purposes only; does not influence ARM model recommendations.
Standard cost (COGS)	The initial cost of producing this product, i.e. the sum of the costs of all materials in the BOM. If multiple configurations or products are considered, use a weighted average cost by volume.

Input field name	Explanation
Selling price (new)	The starting selling price of this product in new condition. If the product is sold with multiple configurations at different prices, use a weighted average price by volume. For products with subsidies, such as smartphones with bounties, include the subsidy. This price can decline over time—see the <i>Selling price erosion</i> field.
Volume sold over lifetime	The total number of units of the product Dell will sell in the specified region over its lifetime, typically based on a sales forecast or production plan.
Selling schedule	The production/sales schedule of the product, which specifies quarter-by-quarter percentages of total volume sold in each quarter of the product's lifetime. There are a number of pre-defined production schedules from which a user can choose, or a user can define his or her own.
Selling price erosion	The depreciation profile to apply to the selling price, which specifies quarter-by-quarter percentage price declines in the selling price of the product as new.
Standard cost erosion	The depreciation profile to apply to the standard cost of the product, which specifies quarter-by-quarter percentage cost declines in COGS.
Inventory holding cost (cost of capital)	Dell's quarterly inventory holding cost, expressed as a percentage of the standard cost of the product.

4.4.1.2. Returns Section

Returns represent the first of Dell's "three Rs". This section is where users indicate anticipated return volumes by return condition/quality. We show this section in Figure 29(b) and describe its input fields in Table 23. As we do in the modeling scenario section, we visualize the user input in a graph for verification—showing quarter-by-quarter return volumes by return condition. Here users have the flexibility of using the inventory types we define—L1, L2, L3, BER, new, DOA, refurbished, and parts—or define their own. We show the interrelationships among these inventory types as they apply to the quality gradient in Figure 28. As a practical matter, if the user chooses to define inventory differently the same quality gradient applies unless the user also changes some of the hard-coded arcs in our underlying network flow formulation.

Table 23: Explanation of input fields in the returns section

Input field name	Explanation
Return rate: L1	The percentage of sales expected to be returned in need of level 1 (light) repair, such as minor cosmetic defects.
Return rate: L2	The percentage of sales expected to be returned in need of level 2 (moderate) repair, such as replacement of inexpensive parts.
Return rate: L3	The percentage of sales expected to be returned in need of level 3 (extensive) repair, such as replacement of expensive parts.
Return rate: BER	The percentage of sales expected to be returned in a state that is beyond economic repair (BER). This applies to cases where the cost of repair (including labor) would exceed the material cost of the device.
Return rate: New	The percentage of sales expected to be returned in as-new condition, such as unopened boxes.
Return rate: DOA	The percentage of sales expected to be returned in dead-on-arrival or defective-on-arrival (DOA) condition. Use when a return-for-exchange agreement is in place with the ODM. If a return-for-credit agreement is in place with the ODM, set to 0%.

Input field name	Explanation
Return rate: Refurbished	The percentage of sales expected to be returned in a refurbished state. This may be an estimate of the percentage of unopened returns, if those returns cannot be sold as new. Otherwise, 0% should typically be used.
Return rate: Parts	The percentage of sales expected to be returned as parts. Generally, this will be 0%.
Same-quarter returns	The proportion of returns that originate from sales in the same quarter. Example: 40% means that if 100 returns are expected from Q1 sales, 40 will be returned in Q1 and 60 in Q2. This percentage applies to L1/L2/L3/BER/New only.
E&O: Refurbished	The percentage of total production volume estimated to become excess & obsolete (E&O) inventory in the form of assembled product.
E&O: Parts	The percentage of total production volume estimated to become excess & obsolete (E&O) inventory in the form of unassembled parts.

4.4.1.3. Rework Section

Rework represents the second of the “three Rs”. This is where users specify which transformational activities they are considering in the reverse supply chain. We show this section in Figure 29(c) and describe its input fields in Table 24. Note that for each transformational activity, the user may specify the type of inventory at each end of the transformation. For example, based on OEM contracts they may have a transformational activity that transforms DOA inventory to new inventory, or to refurbished inventory; they are free to choose which mapping is appropriate, and also define new mappings based on arbitrary inventory types. Our implementation dynamically updates the underlying network flow-based formulation based on the transformational definitions provided.

Table 24: Explanation of input fields in the rework activity section

Input field name	Explanation
Rework method	The name of the transformational activity. The activity itself may be performed by either Dell or a third party.
Inventory: Input type	The inventory type transformed using this rework activity. Users select the inventory type using a drop-down from the inventory types defined in the returns section.
Inventory: Output type	The ending inventory type once the transformation is complete. Here too, users select the inventory type using a drop-down from the inventory types defined in the returns section.
Costs: Cost per box	The activity-based cost (ABC) or variable cost, of transforming one unit.
Costs: Set-up cost	The one-time cost, or fixed cost, of employing this transformational activity. This cost is incurred if and only if one or more units are transformed throughout the lifetime. The amount should not include any sunk costs, which are incurred regardless of whether the transformational activity is utilized.
Capacity: Max devices/qtr	The maximum number of units, per quarter, capable of being transformed using this method. If the field is blank, there is no limit. If the field is “0”, the method is effectively disabled.

4.4.1.4. Recovery Section

Recovery represents the third of Dell’s “three Rs”. This is where users specify the recovery channels they are considering in the reverse supply chain. We show this section in Figure 29(d) and describe

its input fields in Table 25. Note that total revenue earned through any particular channel is based on the sum of three components, inputted separately—fixed revenue, revenue relative to sales price, and revenue relative to COGS.

Table 25: Explanation of input fields in the recovery channel section

Input field name	Explanation
Channel	The name of the recovery channel for informational purposes. The channel may be run by Dell and/or a third party. The edit icon appearing next to the field name, when clicked, takes the user to the Recovery Channel Worksheet screen, where he or she can see itemized revenue and cost component fields for guidance defining the channel.
Inventory: Minimum quality	The minimum quality of inventory required by this channel. As mentioned earlier, the ARM model employs a quality gradient as described in Figure 28, so inventory of higher quality is also considered acceptable.
Costs: Cost per box	The activity-based cost (ABC), or variable cost, of selling one unit through this channel, such as shipping and handling.
Costs: Set-up cost	The one-time cost, or fixed cost, of employing this recovery channel. This cost is incurred if and only if one or more units are ultimately sold through this channel. The amount should not include any sunk costs, which are incurred regardless of whether the recovery channel is utilized.
Revenue: Fixed	Revenue, in absolute terms, earned when one unit is sold through this channel, provided revenue is fixed and does not erode over time; otherwise set to \$0.
Revenue: % of sales price	Revenue, in terms relative to the current sales price, earned when a unit is sold through this channel. The sales price erodes over time per the depreciation profile specified in the selling price erosion field.
Revenue: % of COGS	Revenue, in terms relative to the current standard cost, earned when a unit is sold through this channel. Standard cost erodes over time per the depreciation profile specified in the standard cost erosion field.
Capacity: Max devices/qtr	The maximum number of units per quarter that Dell can sell through this recovery channel. If the field is blank, there is no limit. If the field is "0", the method is effectively disabled.

4.4.2. Production Schedule Worksheet

The ARM model calculates the number of units of inventory type i returned in a quarter t ($w_{i,t}$) using the following formula:

$$w_{i,t} = VR_i P_t S + VR_i P_{t-1} (S - 1) = VR_i (P_t S + P_{t-1} (S - 1))$$

where

V : volume sold over lifetime

R_i : return rate of inventory type i

P_t : production schedule percentage in period t

S : proportion of same-quarter returns

There are two parts of the return: the proportion of units produced in this quarter and returned in this quarter, and the proportion of units produced in the previous quarter and returned in this quarter.

For simplicity, we assume the proportion of sales returned two or more quarters later is 0%; at Dell, the rate is practically zero.

Users specify values for V , R_t , and S directly on the input screen of the ARM model. They choose P_t values for all t by selecting one of the predefined production schedules, editing an existing production schedule, or creating a new one. They use the production schedule worksheet, shown in Figure 30, to do this; in the figure, we show the 10 production schedules pre-defined by the ARM model from which users can choose. The “curved” profiles indicate a typical production lifecycle with a ramp-up period and a ramp-down period. The “flat” profiles are useful for modeling steady-state activity, especially among a group of products or succession of products. We explain the input fields available on this worksheet in Table 26.

Figure 30: Production schedule worksheet

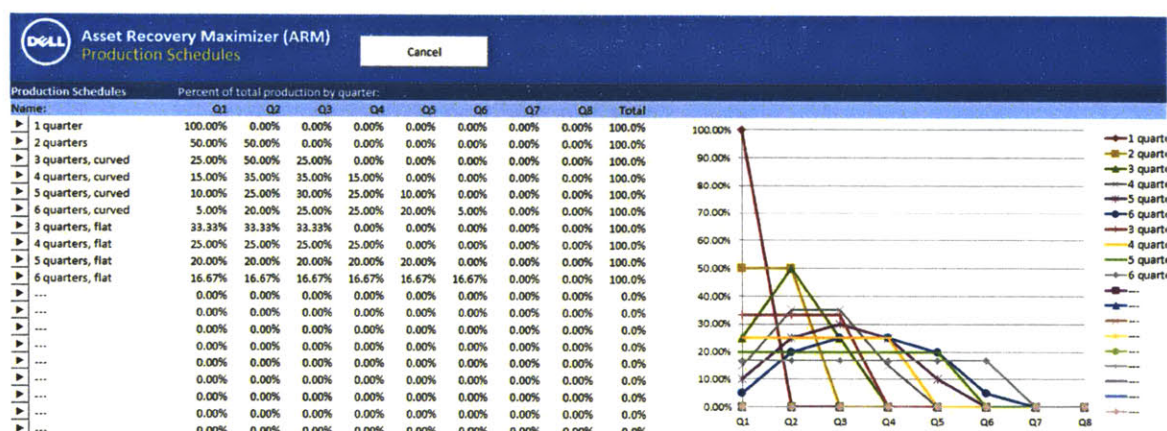


Table 26: Explanation of input fields in the production schedule worksheet

Input field name	Explanation
Profile name	A description to identify the production schedule. The name appears on the input screen as one of the selling schedules from which a user may choose.
Quarterly production percentage	The percentage of the total units produced over the lifetime of the product to be produced/sold in each quarter. The sum of percentages across all quarters should equal 100%.

4.4.3. Depreciation Profiles Worksheet

The depreciation profiles worksheet, as shown in Figure 31, allows users to view, edit, and create depreciation profiles. A *depreciation profile* indicates the rate by which a price or cost erodes in value over time. In Dell’s industry, price erodes chiefly due to technical obsolescence, and standard cost erodes as Dell negotiates better prices with its supplier or receives volume discounts. The two rates are different, so the input screen allows the user to choose a unique profile for each. The 26 profiles pre-defined by the

In addition to net income through recovery channels, cost erosion is a factor in calculating inventory holding cost. The inventory holding cost of inventory of type i at time t ($H_{i,t}$) is calculated straightforwardly as follows:

$$H_{i,t} = h_i c_i$$

We describe the input fields of the depreciation profiles worksheet in Table 27. For informational purposes and data validation, following the input field columns is summary of the total depreciation and a depiction of quarter-by-quarter value changes starting from a basis of 1.00.

Table 27: Explanation of input fields in the depreciation profiles worksheet

Input field name	Explanation
Profile name	A description to identify the depreciation profile. The name appears on the input screen as one of the options from which the user may choose for selling price erosion and for standard cost erosion.
Quarterly depreciation percentage	The proportion of value lost since the last quarter. Thus, the quarterly rates are cumulative. A negative percentage is permissible to express an <i>increase</i> in value.

4.4.4 Recovery Channel Worksheet

Users can describe most recovery channels quite easily on the input screen without the need for this worksheet. This worksheet provides assistance in determining the revenue and cost values of a recovery channel input row under for channels with complex revenue models and/or costs. We show an example of such a channel in Figure 32. For this channel, we see multiple sources of revenue, including a flat price, a revenue share arrangement, an activation bounty paid by the telecommunications carrier with an amount that varies by plan type. There are also two cost components: shipping and a transaction fee. Finally, there is a \$50 Dell Outlet coupon offered as an extra incentive to purchase, which is deducted from revenue at an anticipated redemption rate. In accordance with Dell accounting practices, we treat the coupon as negative revenue (contra-revenue) instead of a cost. This complex example illustrates the occasional need for this worksheet.

The worksheet has four sections: channel profile, revenue per device, cost per device, and channel results. We discuss each of these in turn.

Figure 32: Recovery channel worksheet

Asset Recovery Maximizer (ARM)
Recovery Channel Worksheet

Apply Changes

Cancel

Channel Profile

Channel name	Rev share with smartphone activation	
Maximum volume	5,000	devices per quarter (or leave blank for no maximum)
One-time set-up cost	\$0.00	to enable this channel
Inventory type	Refurbished	quality or better

Revenue per Device		Rev share stake:	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Fixed price	\$100.00	100.00%	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00
Relative revenue: % of COGS	0.00%	100.00%	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Relative revenue: % of sales price	10.00%	100.00%	\$55.00	\$48.26	\$42.35	\$37.16	\$32.61	\$28.62	\$25.11	\$22.04
Plus bounties:		Bounty:	Frequency:							
New activation	\$400.00	25.00%	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00	\$100.00
New family plan activation	\$250.00	25.00%	\$62.50	\$62.50	\$62.50	\$62.50	\$62.50	\$62.50	\$62.50	\$62.50
Upgrade	\$300.00	50.00%	\$150.00	\$150.00	\$150.00	\$150.00	\$150.00	\$150.00	\$150.00	\$150.00
Average bounty	\$312.50	100.00%								
Less rebates:		Redemption rate:								
Dell.com coupon	\$0.00	0.00%	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Dell Outlet coupon	\$50.00	50.00%	\$25.00	\$25.00	\$25.00	\$25.00	\$25.00	\$25.00	\$25.00	\$25.00
Other rebate	\$0.00	0.00%	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Expected redemption value	\$25.00									
Total revenue per device	Varies by quarter		\$442.50	\$435.76	\$429.85	\$424.66	\$420.11	\$416.12	\$412.61	\$409.54

Cost per Device		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Shipping to channel:	\$5.00	\$5.00	\$5.00	\$5.00	\$5.00	\$5.00	\$5.00	\$5.00	\$5.00
Shipping to customer:	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Fixed stocking/holding cost	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Transaction fee	\$50.00	\$50.00	\$50.00	\$50.00	\$50.00	\$50.00	\$50.00	\$50.00	\$50.00
Other fees	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Total cost per device	\$55.00	\$55.00	\$55.00	\$55.00	\$55.00	\$55.00	\$55.00	\$55.00	\$55.00

Channel Results		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Net revenue per device	Varies by quarter	\$387.50	\$380.76	\$374.85	\$369.66	\$365.11	\$361.12	\$357.61	\$354.54

4.4.4.1. Channel Profile Section

The channel profile section allows a user to user specify general information about the channel, exactly as it appears on the channel's row on the input screen. See Table 25 for a description of the input fields in this section.

4.4.4.2. Revenue per Device Section

The revenue per device section breaks down of the revenue components related to selling inventory through a recovery channel. See Table 28 for a description of the input fields in this section.

Table 28: Explanation of input fields in the revenue per device section of the recovery channel worksheet

Input field name	Explanation
Fixed price	If the product is sold through this channel at a fixed price—that does not diminish over time—specify it here. Otherwise specify 0.
Relative revenue: % of COGS	If the product is sold through this channel at a percentage of the product's standard cost, specify it here. Otherwise specify 0. If, on the <i>Inputs</i> worksheet, you have selected a standard cost erosion depreciation profile, the revenue will decline over time in accordance with that profile. Use the quarter-by-quarter revenue grid to the right of this field to verify the product is priced at the value you expect.
Relative revenue: % of sales price	Indicates the percentage of the product's as-new sales price for which it is sold through this channel, if pricing is relative to the as-new price. Otherwise specify 0.
Rev share stake: Fixed price	The percentage of the fixed price to be earned by Dell. This should be set to 100% if there is no revenue share with a third party.
Rev share stake: % of COGS	The percentage of the COGS-based revenue to be earned by Dell. This should be set to 100% if there is no revenue share with a third party.
Rev share stake: % of sales price	The percentage of the sales price-based revenue to be earned by Dell. This should be set to 100% if there is no revenue share with a third party.
Bounty: New activation	For devices that use an optional or required wireless carrier plan, such as netbooks and smartphones, the earnings Dell will receive from the carrier for activating a single new customer on an individual plan.
Bounty: Family plan	For devices that use an optional or required wireless carrier plan, such as netbooks and smartphones, indicate the earnings Dell will receive from the carrier for activating a new customer on a family plan.
Bounty: Upgrade	For devices that use an optional or required wireless carrier plan, such as netbooks and smartphones, indicate the earnings Dell will receive from the carrier for upgrading/renewing a customer's existing plan.
Frequency: New activation	The proportion of customers who sign up for a new individual plans. The sum of this and the other frequency estimates must sum to no more than 100%.
Frequency: Family plan	The proportion of customers who will sign up for new family plans. The sum of this and the other frequency estimates must sum to no more than 100%.
Frequency: Upgrade	The proportion of customers who will upgrade/renew their existing plans. The sum of this and the other frequency estimates must sum to no more than 100%.
Dell.com coupon	The value of the Dell.com coupon to be given to consumers who purchase the product through this channel. Specify \$0 if no coupon is being offered.
Dell Outlet coupon	The value of the Dell Output coupon to be given to consumers who purchase the product through this channel. Specify \$0 if no coupon is being offered.
Other rebate	The value of all other rebates to be given to consumers who purchase the product through this channel. Specify \$0 if no other rebate is being offered.
Redemption rate: Dell.com coupon	The percentage of consumers who are likely to redeem the Dell.com coupons they receive with purchase.
Redemption rate: Dell Outlet coupon	The percentage of consumers who are likely to redeem the Dell Outlet coupons they receive with purchase.
Redemption rate: Other rebate	The percentage of consumers who are likely to redeem any other rebate offers they receive with purchase.

4.4.4.3 Cost per Device Section

The cost per device section breaks down the cost components of selling inventory through a recovery channel. See Table 29 for a description of the input fields in this section.

Table 29: Explanation of input fields in the cost per device section of the recovery channel worksheet

Shipping to channel	The cost of shipping and handling a single unit to this recovery channel for stocking, before customer purchase, if Dell pays this cost. If the unit is shipped directly to the customer, this cost may be \$0.
Shipping to customer	The cost of shipping and handling a single unit from the recovery channel to the customer, if Dell pays this cost.
Fixed stocking/holding cost	The cost of stocking or holding inventory in this channel, if Dell pays this cost. This is additional cost above the standard cost-based inventory holding cost the ARM automatically calculates for unsold WIP or finished goods.
Transaction fee	The cost to Dell of any transaction fee related to the processing and sale of one unit of product.
Other fees	Any other costs to Dell associated with the processing and sale of one unit of product.

4.4.4.4. Channel Results Section

The channel results section summarizes net income, per unit, in the channel, excluding fixed costs and rework expenses that went into the inventory (if any). If the net income varies by quarter due to price erosion or cost erosion, the net revenue is reported as “varies by quarter.” In this case, as shown in Figure 32, quarter-by-quarter revenue-per-unit numbers reported alongside this message indicate how per-unit net income changes over time in this channel.

4.5 Recommendation Engine

In this section, we describe the three worksheets that constitute the recommendation engine upon which the ARM model’s outputs are based: the nodes worksheet, formulation worksheet, and cost attributions worksheet. We describe each of these worksheets in turn.


4.5.1. Nodes Worksheet

The nodes worksheet, the top portion of which we show in Figure 33, defines the nodes in the formulation’s flow network. It tracks the inflows, outflows, and current inventory levels of each node as well as the associated income and expenses (save fixed costs) over all periods. Each node has one of the following classifications:

- **Inventory nodes:** These nodes, identified by numbers 1-8, represent returned inventory of a particular quality, such as BER or unopened/new. Units may move from one inventory node to another through transformational activities or the downgrade process.
- **Channel nodes:** These nodes, identified by numbers 20-26, represent recovery channels. Units moved into one of these channels represents a sale to the channel.

- Unassigned nodes: These nodes, identified by numbers 998-999, are reserved for future use and not currently part of the flow network. We include them now to make it easy for maintainers of the model to extend its capabilities with a minimum number of changes.

Figure 33: Top portion of the nodes worksheet, where we define the 17 nodes in the flow network

 Asset Recovery Maximizer (ARM) Network Definition								
Prices & Costs	1	2	3	4	5	6	7	8
Relative price	\$1.00	\$0.88	\$0.77	\$0.68	\$0.59	\$0.52	\$0.46	\$0.40
Selling price	\$125.00	\$109.69	\$96.26	\$84.47	\$74.12	\$65.04	\$57.08	\$50.09
Relative cost	\$1.00	\$0.97	\$0.94	\$0.91	\$0.88	\$0.85	\$0.82	\$0.80
Standard cost (COGS)	\$100.00	\$96.80	\$93.70	\$90.70	\$87.80	\$84.98	\$82.26	\$79.63
LCM	\$100.00	\$96.80	\$93.70	\$84.47	\$74.12	\$65.04	\$57.08	\$50.09

Node Properties				Revenues		Costs	
Node ID	Category	Name	Holding cost?	Fixed	% of price	% of COGS	Cost
1	Inventory	L1	1	\$0.00	0.00%	0.00%	0.00
2	Inventory	L2	1	\$0.00	0.00%	0.00%	0.00
3	Inventory	L3	1	\$0.00	0.00%	0.00%	0.00
4	Inventory	BER	1	\$0.00	0.00%	0.00%	0.00
5	Inventory	New	1	\$0.00	0.00%	0.00%	0.00
6	Inventory	DOA	1	\$0.00	0.00%	0.00%	0.00
7	Inventory	Refurbished	1	\$0.00	0.00%	0.00%	0.00
8	Inventory	Parts	1	\$0.00	0.00%	0.00%	0.00
20	Channel	Dell.com	0	\$0.00	100.00%	0.00%	5.00
21	Channel	Dell Outlet	0	\$0.00	85.00%	0.00%	5.00
22	Channel	Auction	0	\$0.00	0.00%	60.00%	5.00
23	Channel	Parts	0	\$0.00	0.00%	30.00%	5.00
24	Channel	---	0	\$0.00	0.00%	0.00%	0.00
25	Channel	---	0	\$0.00	0.00%	0.00%	0.00
26	Channel	---	0	\$0.00	0.00%	0.00%	0.00
998	---	---	0	\$0.00	0.00%	0.00%	0.00
999	---	---	0	\$0.00	0.00%	0.00%	0.00

In Figure 34, we show the nodes worksheet in its entirety. There are 10 sections where calculations progress, from definitions to inflows, inflows to outflows, outflows to ending stock levels, and stock levels to income, expenses, and net income. We describe these sections in more detail in Table 30.

Figure 34: The network flow worksheet in its entirety, with all ten sections (a) through (j) identified.

Table 30: Descriptions of the sections in the nodes worksheet

Section	Description
Prices & costs: Figure 34(a)	Calculates product price and cost information period-by-period, with price and cost erosion taken into account.
Node properties: Figure 34(b)	Defines the nodes in the flow network, whether inventory at the node incurs holding costs at each node, and per-unit revenue and cost information at each node.
Return streams: Figure 34(c)	Calculates the incoming volume of returned product at each node in each period.
Return streams rounded: Figure 34(d)	Rounds the return stream volumes to whole numbers to leverage the integrality property of flow networks, as discussed previously.
Inflows: Figure 34(e)	Indicates the total volume of product arriving at each node in each period; this is the sum of new returns plus inflow from other nodes.
Outflows: Figure 34(f)	Indicates the total volume of product departing each node in each period; this is outflow to other nodes.
Ending stock: Figure 34(g)	Indicates the ending inventory level at the end of each period, defined as starting stock plus inflows minus outflows.
Node income from inflows: Figure 34(h)	Calculates the revenue earned from inflows to each channel node in each period.
Node expenses: Figure 34(i)	Calculates the expenses associated with inflows to each channel node in each period.
Node net income: Figure 34(j)	Calculates the net income of flows to each channel node in each period, defined as income less expenses.

4.5.2. Formulation Worksheet

The formulation worksheet is where we implement the model formulation described in Section 4.2. We show this sheet and identify its five constituent sections in Figure 35, which we describe in Table 31.

Figure 35: Formulation worksheet with its five sections: (a) objective function, (b) flow decision variables, (c) rework and channel inventory decision variables, (d) capacity constraints, and (e) flow constraints




Dell Asset Recovery Maximizer (ARM) Formulation										
Last calculated on 12/01/2010 at 00:39:28										
Objective Function										
Objective: Maximize net income										
		Recovery income	\$3,915,396							
		Less recovery variable expenses	\$56,000							
		Less recovery set-up expenses	\$0							
		Less rework variable expenses	\$325,500							
		Less rework set-up expenses	\$0							
		Net income	\$3,533,896							
		Net recovery	95.4%							
Flow Decision Variables										
From	To	1	2	3	4	5	6	7	8	Total
1	7 L1 -> Refurbished	1,050	1,050	1,050	1,050	-	-	-	-	4,200
1	4 L1 -> BER	-	-	-	-	-	-	-	-	-
2	7 L2 -> Refurbished	1,050	1,050	1,050	1,050	-	-	-	-	4,200
2	4 L2 -> BER	-	-	-	-	-	-	-	-	-
3	7 L3 -> Refurbished	-	-	-	-	-	-	-	-	-
3	4 L3 -> BER	-	-	-	-	-	-	-	-	-
6	5 DOA -> New	525	525	525	525	-	-	-	-	2,100
6	4 DOA -> BER	-	-	-	-	-	-	-	-	-
5	7 New -> Refurbished	569	569	569	569	-	-	-	-	2,276
7	4 Refurbished -> BER	-	-	-	-	-	-	-	-	-
4	8 BER -> Parts	-	-	-	-	-	-	-	-	-
7	20 Refurbished -> Services	-	-	-	-	-	-	-	-	-
7	21 Refurbished -> Seed stock	-	-	-	-	-	-	-	-	-
7	22 Refurbished -> Simplicity / Outlet	2,669	2,669	2,669	2,669	-	-	-	-	10,676
4	23 BER -> Auction	131	131	131	131	-	-	-	-	524
4	24 BER -> ATCIE auction	-	-	-	-	-	-	-	-	-
4	25 BER -> S+ technologies	-	-	-	-	-	-	-	-	-
8	26 Parts -> Part harvesting	-	-	-	-	-	-	-	-	-
Rework & Channel Binary Decision Variables (and "Big M" Constraints)										
Which rework options and recovery channels should be used? These decision variables are bound by "big M" constraints.										
		Use it?	Volume	Set up costs	Var costs	Big M				
Rework:	Level 1 repair	✓ 1	4,200	\$0.00	\$25.00	✓ 9,999,999				
	Level 2 repair	✓ 1	4,200	\$0.00	\$50.00	✓ 9,999,999				
	Level 3 repair	✗ 0	-	\$0.00	\$9,999.00	✓ 9,999,999				
	Downgrade L1	✗ 0	-	\$0.00	\$0.00	✓ 9,999,999				
	Downgrade L2	✗ 0	-	\$0.00	\$0.00	✓ 9,999,999				
	Downgrade L3	✗ 0	-	\$0.00	\$0.00	✓ 9,999,999				
	Downgrade new	✓ 1	2,276	\$0.00	\$0.00	✓ 9,999,999				
	ODM Exchange	✓ 1	2,100	\$0.00	\$5.00	✓ 9,999,999				
	Teardown	✗ 0	-	\$0.00	\$9,999.00	✓ 9,999,999				
Recovery:	Services	✗ 0	-	\$0.00	\$5.00	✓ 9,999,999				
	Seed stock	✗ 0	-	\$0.00	\$5.00	✓ 9,999,999				
	Simplicity / Outlet	✓ 1	10,676	\$0.00	\$5.00	✓ 9,999,999				
	Auction	✓ 1	524	\$0.00	\$5.00	✓ 9,999,999				
	ATCIE auction	✗ 0	-	\$0.00	\$5.00	✓ 9,999,999				
	S+ technologies	✗ 0	-	\$0.00	\$5.00	✓ 9,999,999				
	Part harvesting	✗ 0	-	\$0.00	\$5.00	✓ 9,999,999				
Capacity Constraints										
Ensures no rework option or recovery channel is used beyond capacity.										
		1	2	3	4	5	6	7	8	Max Capacity
Rework:	Level 1 repair	✓	✓	✓	✓	✓	9,999,999	9,999,999	9,999,999	9,999,999
	Level 2 repair	✓	✓	✓	✓	✓	9,999,999	9,999,999	9,999,999	9,999,999
	ODM Exchange	✓	✓	✓	✓	✓	1,000	1,000	1,000	1,000
	Teardown	✓	✓	✓	✓	✓	9,999,999	9,999,999	9,999,999	9,999,999
Recovery:	Services	✓	✓	✓	✓	✓	350	350	350	350
	Seed stock	✓	✓	✓	✓	✓	700	700	700	700
	Simplicity / Outlet	✓	✓	✓	✓	✓	9,999,999	9,999,999	9,999,999	9,999,999
	Auction	✓	✓	✓	✓	✓	9,999,999	9,999,999	9,999,999	9,999,999
	ATCIE auction	✗	✗	✗	✗	✗	-	-	-	-
	S+ technologies	✗	✗	✗	✗	✗	-	-	-	-
	Part harvesting	✗	✗	✗	✗	✗	-	-	-	-
Flow Constraints										
Ensures only positive quantities flow from a node, and no inv left at end.										
		1	2	3	4	5	6	7	8	
1		✓	✓	✓	✓	✓	0	0	0	0
2		✓	✓	✓	✓	✓	0	0	0	0
3		✓	✓	✓	✓	✓	0	0	0	0
4		✓	✓	✓	✓	✓	0	0	0	0
5		✓	✓	✓	✓	✓	0	0	0	0
6		✓	✓	✓	✓	✓	0	0	0	0
7		✓	✓	✓	✓	✓	0	0	0	0
8		✓	✓	✓	✓	✓	0	0	0	0

Table 31: Descriptions of the sections in the formulation worksheet

Section	Description
Objective function: Figure 35(a)	Calculates the formulation's objective function as well as related financial measures, including net recovery
Flow decision variables: Figure 35(b)	Dynamically constructs the 18 arcs of the flow network based on user input. For each arc, provides a series of decision variables to track product flow in each period.
Rework and channel inventory decision variables: Figure 35(c)	Defines the 16 binary decision variables to enable/disable the transformational activities and recovery channels, along with the associated big-M constraints and volume and cost calculations.
Capacity constraint: Figure 35(d)	Defines all capacity constraints.
Flow constraints: Figure 35(e)	Defines all flow constraints.

In the implementation, we use icon circular indicators to identify cells with constraints. The color and inner content of each icon reveals whether the constraint is currently being held, as shown in Table 38.

Table 32: Explanations of constraint status icons

Constraint icon	Meaning
	The constraint is being held; all decision variables are within acceptable range.
	The constraint is being violated within a very small margin of error, typically less than 0.001. All decision variables are still considered to be within acceptable range.
	The constraint is being violated. One or more decision variables are not within an acceptable range.

4.5.3. Cost Attributions Worksheet

The cost attributions worksheet, shown in Figure 36, determines the total cost of each recovery channel to calculate the channels' respective net recovery metrics. While each recovery channel has both a cost per box and set-up cost, these are not the only costs; each unit sold through the channel is burdened with its own costs if it underwent one or more transformational activities. This worksheet takes those burdened costs into consideration and distributes them to the appropriate recovery channels. The multi-step calculation process proceeds through the eight sections of the worksheet, which we define in Table 33.

Figure 36: Cost attributions worksheet with its eight sections identified: (a) recovery channel costs, (b) recovery channel rework volumes, (c) recovery channel rework proportions, (d) inventory to channel vs. inventory sources, (e) rework enabled, (f) rework cost per box, (g) rework setup costs, and (h) total costs

D4LL Asset Recovery Maximizer (ARM) Cost Attributions									
(a)	Recovery channel costs	Channel income	Channel cost per box	Channel setup cost	Rework cost	COGS	Net income	Net recovery	
	Services	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00		
	Seed stock	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00		
	Simplicity / Outlet	\$3,880,567.95	\$51,380.00	\$0.00	\$315,000.00	\$3,547,947.84	\$3,512,187.95	99.1%	
	Auction	\$34,828.11	\$2,620.00	\$0.00	\$0.00	\$174,140.56	\$32,208.11	19.7%	
	ATCUE Auction	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00		
	5+ technologies	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00		
	Part harvesting	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00		
(b)	Recovery channel rework volumes	BER	New	DOA	Refurbished	Parts			
	Services	0	0	0	0	0			
	Seed stock	0	0	0	0	0			
	Simplicity / Outlet	0	0	0	10674	0			
	Auction	134	0	0	0	0			
	ATCUE Auction	0	0	0	0	0			
	5+ technologies	0	0	0	0	0			
	Part harvesting	0	0	0	0	0			
(c)	Recovery channel rework proportions	BER	New	DOA	Refurbished	Parts			
	Services	0.0%	0.0%	0.0%	0.0%	0.0%			
	Seed stock	0.0%	0.0%	0.0%	0.0%	0.0%			
	Simplicity / Outlet	0.0%	0.0%	0.0%	100.0%	0.0%			
	Auction	100.0%	0.0%	0.0%	0.0%	0.0%			
	ATCUE Auction	0.0%	0.0%	0.0%	0.0%	0.0%			
	5+ technologies	0.0%	0.0%	0.0%	0.0%	0.0%			
	Part harvesting	0.0%	0.0%	0.0%	0.0%	0.0%			
	Total cost:	\$0.00	\$10,300.00	\$0.00	\$315,000.00	\$0.00			
(d)	Inventory to channel \ inventory sources	L1	L2	L3	BER	New	DOA	Refurbished	Parts
	L1	0	0	0	0	0	0	0	4200
	L2	0	0	0	0	0	0	0	4200
	L3	0	0	0	0	0	0	0	0
	BER	0	0	0	0	0	0	0	134
	New	0	0	0	0	2100	0	0	2276
	DOA	0	0	0	0	0	0	0	2100
	Refurbished	4200	4200	0	0	2276	0	0	10676
	Parts	0	0	0	0	0	0	0	0
(e)	Rework enabled	L1	L2	L3	BER	New	DOA	Refurbished	Parts
	L1	0	0	0	0	0	0	0	0
	L2	0	0	0	0	0	0	0	0
	L3	0	0	0	0	0	0	0	0
	BER	0	0	0	0	0	0	0	0
	New	0	0	0	0	0	1	0	0
	DOA	0	0	0	0	0	0	0	0
	Refurbished	1	1	0	0	1	0	0	0
	Parts	0	0	0	0	0	0	0	0
(f)	Rework cost per box	L1	L2	L3	BER	New	DOA	Refurbished	Parts
	L1	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L2	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L3	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	BER	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	New	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$5.00	\$0.00	\$0.00
	DOA	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	Refurbished	\$25.00	\$50.00	\$9,999.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10,674.00
	Parts	\$0.00	\$0.00	\$0.00	\$9,999.00	\$0.00	\$0.00	\$0.00	\$9,999.00
(g)	Rework setup costs	L1	L2	L3	BER	New	DOA	Refurbished	Parts
	L1	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L2	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L3	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	BER	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	New	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	DOA	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	Refurbished	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	Parts	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
(h)	Total costs	L1	L2	L3	BER	New	DOA	Refurbished	Parts
	L1	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L2	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	L3	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	BER	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	New	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$10,300.00	\$0.00	\$10,300.00
	DOA	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
	Refurbished	\$105,000.00	\$210,000.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$315,000.00
	Parts	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00

Table 33: Descriptions of the sections in the cost attributions worksheet

Section	Description
Recovery channel costs: Figure 36(a)	Provides a full accounting of each recovery channel's costs and associated net recovery: channel income, channel cost-per-box, channel set-up cost, rework cost, total COGS, net income, and net recovery. The calculations are straightforward, except for rework cost. The subsequent section provide the intermediate calculations to arrive at this cost.
Recovery channel rework volumes: Figure 36(b)	Indicates the total volume of inventory of each type, across all quarters, ultimately sold through each recovery channel.

Section	Description
Recovery channel rework proportions: Figure 36(c)	Based on the data in the previous section, indicates the percentage of inventory of each type sold through each recovery channel.
Inventory to channel vs. inventory sources: Figure 36(d)	Indicates the number of units transformed from each inventory type to all others.
Rework enabled: Figure 36(e)	Indicates which of the inventory transformations are defined in the network based on user input. A 1 indicates the transformation is defined; a 0 means it is not. For example, a DOA-to-new transformation may be defined, whereas an L1-to-new may not be.
Rework cost per box: Figure 36(f)	Indicates the cost per box, as specified on the Inputs worksheet, for each of the enabled rework activities.
Rework setup costs: Figure 36(g)	Indicates the set-up cost, as specified on the Inputs worksheet, for each of the enabled rework activities.
Total costs: Figure 36(h)	Indicates the total cost (the product of cost-per-box and the number of boxes, plus any set-up cost) for each of the enabled rework activities.

4.6. Outputs

In this section, we describe the three worksheets that provide the user with details results: the recommendations worksheet, network flow worksheet, and sensitivity analysis worksheet. We describe each of these worksheets in turn.

4.6.1. Recommendations Worksheet

The recommendations worksheet, shown in Figure 37, summarizes the ARM Model's recommended reverse supply chain design. This worksheet appears to users after they use the input screen to specify inputs and click the "Generate Recommendations" button to run the branch-and-bound algorithm on the network flow formulation to find the optimal design. Recommendations are divided into five sections, which we describe in the following subsections: rework activities, recovery channels, implementation plan, financial summary, and benefit-per-box.

Figure 37: Recommendations worksheet. The five sections are (a) rework activities, (b) recovery channels, (c) implementation plan, (d) financial summary, and (e) benefit per box.



4.6.1.1. Rework Activities Section

This section provides a go/no-go decision for each rework activity, along with estimates for volume, net recovery, and utilization. We describe its columns in Table 34.

Table 34: Explanation of columns in the rework activities section

Output column name	Explanation
Recommendation	Indicates whether the rework activity should be employed. The recommendation will be one of the following: <ul style="list-style-type: none"> Enable: Perform this rework activity Do not enable: Do not perform this rework activity Disabled: The rework activity cannot be used because its maximum capacity is set at zero on the input screen.
Volume	Indicates the expected number of units to be reworked using this method throughout the lifetime of this product.
Volume trend	Indicates how the expected number of units reworked via this method will change over time.
Utilization	Indicates how close to capacity this rework method is estimated to be at peak volume. If no capacity is specified, the utilization is reported as 0%.
Cost per box	Indicates the cost per box for this rework activity, accounting for any set-up costs.

4.6.1.2. Recovery Channels Section

This section provides go/no-go decisions for each recovery channel, along with estimates for volume, net recovery, and utilization. We describe its columns in Table 35.

Table 35: Explanation of output columns in the recovery channels section

Output column name	Explanation
Recommendation	Indicates whether this recovery channel should be used to sell inventory. The recommendation will be one of the following: <ul style="list-style-type: none"> • Enable: Use this recovery channel • Do not enable: Do not use this recovery channel • Disabled: The recovery channel cannot be used because its maximum capacity is set at zero on the input screen.
Volume	Indicates the expected number of units to be sold through this recovery channel throughout the lifetime of the product.
Net recovery	Estimates net recovery obtained through this channel, where net recovery is defined as income divided by the sum of COGS and expenses.
Utilization	Indicates how close to capacity this recovery channel is estimated to be at peak volume. If no capacity is specified, the utilization is reported as 0%.
Cost per box	Indicates the cost per box for this recovery channel, accounting for both set-up costs and rework costs.

4.6.1.3. Implementation Plan Section

Through our usability testing, we learned that users sought a way to relate the strategic recommendations in the rework and recovery sections to a tactical implementation they could apply to the plant floor. Based on that feedback, we provide the implementation plan section, which summarizes how reverse supply chain managers should implement the recommendations. For each quality level of inventory, this section describes, proportionally, which transformation activities to perform on the inventory. We explain these actions in Table 36. In this section, we base all recommendations on volume over the *entire* life of the product. The specific percentages may vary by quarter as return rates and price/cost dynamics change, however; for quarter-by-quarter details, users may consult the formulation worksheet.

Table 36: Implementation recommendations

Implementation action	Explanation
% to repair	For a particular type of inventory, indicates the overall percentage that should be repaired.
% to teardown	For a particular type of inventory, indicates the overall percentage that should be torn down.
% to ODM exchange	For a particular type of inventory, indicates the overall percentage that should be sent back to the ODM for exchange.

4.6.1.4. Financial Summary Section

The financial summary section provides income and expense estimates, along with overall net income and net recovery metrics. We summarize these output fields in Table 37. Note that we calculate

net recovery two ways: 1) based on the standard cost model, which Dell's accounting group prefers, and 2) based on the LCM model.

Table 37: Financial summary fields

Income	Indicates the total earnings obtained by selling units through all enabled recovery channels throughout the life of the product.
Rework: Variable expenses	Indicates the total per-box expenses related to rework activities.
Rework: Set-up expenses	Indicates the total set-up expenses related to rework activities. This will equal the sum of the set-up costs for each enabled rework activity. The ARM model assumes set-up costs need not be paid for rework activities that are not enabled.
Recovery: Variable expenses	Indicates the total per-box expenses related to recovery channel activities.
Recovery: Set-up expenses	Indicates the total set-up expenses related to recovery channels. This will equal the sum of the set-up costs for each enabled rework activity. The ARM model assumes set-up costs need not be paid for recovery channels that are not enabled.
Expenses	Provides the total of all expenses: rework variable expenses, rework set-up expenses, recovery variable expenses, and recovery set-up expenses.
Net income	Subtracts all expenses from income to determine the overall contribution to Dell. This number represents Dell's earnings from the reverse supply chain rework and recovery activities. Without the reverse supply chain, this number would be 0, so, in effect, this number measures the impact the reverse supply chain has on Dell's bottom line. The ARM model is designed to recommend a reverse supply chain design that gets net income as high as possible. Net income does <i>not</i> account for the cost of goods returned (COGR), as COGR is a "sunk cost"—something Dell incurs whether or not a reverse supply chain is set up.
Standard COGR	Indicates the overall cost of goods returned (COGR) throughout the lifetime of the product. The number is based on the standard cost of the units returned. If you have specified a standard cost depreciation profile on the <i>Inputs</i> sheet, the standard cost of each returned unit will be based on the standard cost in the quarter in which is what returned, based on that profile.
Net recovery	Indicates the proportion of <i>Standard COGR</i> that is reclaimed by the reverse supply chain. The ARM model calculates this as <i>Net income</i> divided by <i>Standard COGR</i> . This calculation is consistent with Asset Recovery Business' (ARB's) definition of net recovery.
LCM COGR	Provides an alternative view of cost of goods returned (COGR) based on the principle of lower-cost-or-market (LCM). In quarters where the selling price is lower than standard cost, ARM will value the return at the selling price when calculating LCM COGR, whereas for Standard COGR it will always use standard price. Therefore this number will always be equal to or lower than Standard COGR. This number may provide a clearer representation of the value of returned product, though Dell accounting does <i>not</i> use this metric today, favoring Standard COGR instead.
LCM net recovery	Provides an alternative view of net recovery based on LCM COGR instead of Standard COGR. The ARM model calculates this as net income divided by LCM COGR. LCM net recovery will always be greater than or equal to (standard) net recovery. It provides a clearer representation of the true value of the reverse supply chain, though Dell accounting does not use this metric today, favoring (standard) net recovery instead.

4.6.1.5. Benefit-per-box Column

As shown in Figure 37(e), we provide two additional columns in the rework activities and recovery channels section: 1) benefit per box, and 2) net benefit. *Net benefit* answers a what-if question for

act activity or channel. If the channel is enabled, the question is—what benefit does this activity/channel provide? If the channel is disabled, the question is—how much does the price have to decrease for this channel to be used? We add some precision to this definition in Table 38. Benefit per box is simply the ratio of net benefit to the total volume of returns.

Table 38: Explanation of benefit-per-box calculations based on the current recommendation for the rework activity or recovery channel

Current recommendation	Meaning of <i>benefit per box</i>	How benefit per box is calculated
Enable	Indicates the additional cost per box over the entire volume of returned product if this particular rework activity (or recovery channel) were disabled.	Difference in net income between two reverse supply chain designs: an alternative state with the rework activity (or recovery channel) enabled, and the current state with it enabled
Do not enable	Indicates the savings per box over the entire volume of return product if this particular rework activity (or recovery channel) were available at zero cost: that is, a cost per box of \$0 and a set-up cost of \$0. Benefit per box will be \$0 if, even offered at zero cost, the activity/channel is still not used, or, in the case of a recovery channel, is used by offers no revenue opportunity.	Difference in net income between two reverse supply chain designs: an alternative state where the rework activity (or recovery channel) is offered at zero cost, and the current state with it unused.
Disabled	Indicates the savings per box over the entire volume of returned product if this rework activity (or recovery channel) were not disabled.	Difference in net income from two reverse supply chain designs: one with the rework activity (or recovery channel) enabled, and the current state with its capacity limited to zero units.

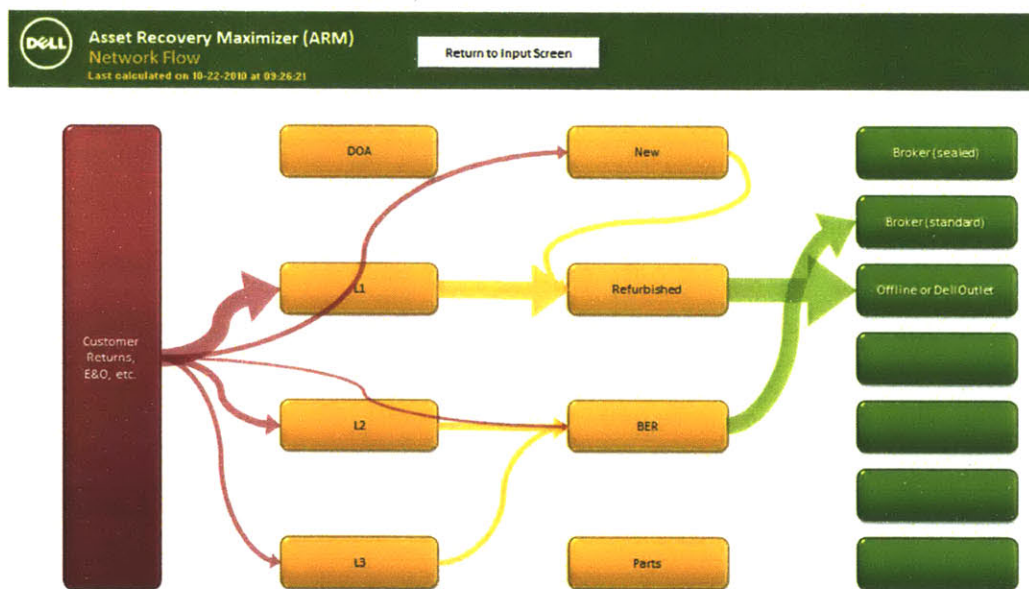
Net benefit calculations are especially helpful in determining how much the per-box costs associated with the channel may rise before alternative activities or channels are preferred. The ARM model does not perform benefit-per-box automatically because doing so is time-consuming (about 15 seconds) and the results are not necessary for typical analyses. Users may click the “Calculate” button (shown in Figure 37(e)) to generate the calculations, which are performed by re-running the optimization algorithm with progressive rework activities and recovery channels either disabled or enabled. We list the macro implementing benefit-per-box calculations in Section B.3.

In certain cases, the ARM model will be unable to compute a net benefit. This happens when the rework activity is currently enabled and must continue to be enabled. If it were disabled, the formulation would not have a feasible solution because not all units flow could through the reverse supply chain, either because 1) none of the available recovery channels is able to sell existing inventory of a given type (such as new, refurbished, BER, etc.), or 2) the channels that can sell this inventory are capacity constrained, and there is more inventory than capacity. In these cases, instead of a net benefit value, users see the message “Required.”

4.6.2. Network Flow Worksheet

Using a network flow diagram as pictured in Figure 38, the network flow worksheet visualizes the decisions and volume estimates shown on the recommendations worksheet. Product returns flow from left to right through the “three Rs” of the reverse supply chain, as defined on the input screen. The thicknesses of the arrows indicate relative volume of product flowing through that path over the lifetime of the product. The thicker the arrow, the more product ARM estimates will be flowing through that path. We describe the macro that generates this diagram in Section B.2.

Figure 38: Dynamic network flow diagram of an example reverse supply chain. Yellow nodes represent inventory types, green nodes represent recovery channels, arrows (arcs) represent enabled transformational activities, and arrow widths convey relative flow volumes.



4.6.3. Sensitivity Analysis

The sensitivity analysis worksheet, shown in Figure 39, is useful in a number of contexts, as we explore through case studies in subsequent chapters. Its two main goals are:

1. To account for uncertainty: Instead of picking a specific value for an input like price erosion, user can specify a range of possible values.
2. To perform multivariate analysis: Users can see how recommendations might change as multiple inputs change, both in tabular form (as an Excel PivotTable) and graphical form (as an Excel PivotChart).

We divide the worksheet into five sections, which we summarize in Table 39.

Figure 39: Sensitivity analysis worksheet, consisting of (a) scenarios, (b) PivotChart, (c) data table, (d) PivotTable, and (e) PivotTable sidebar

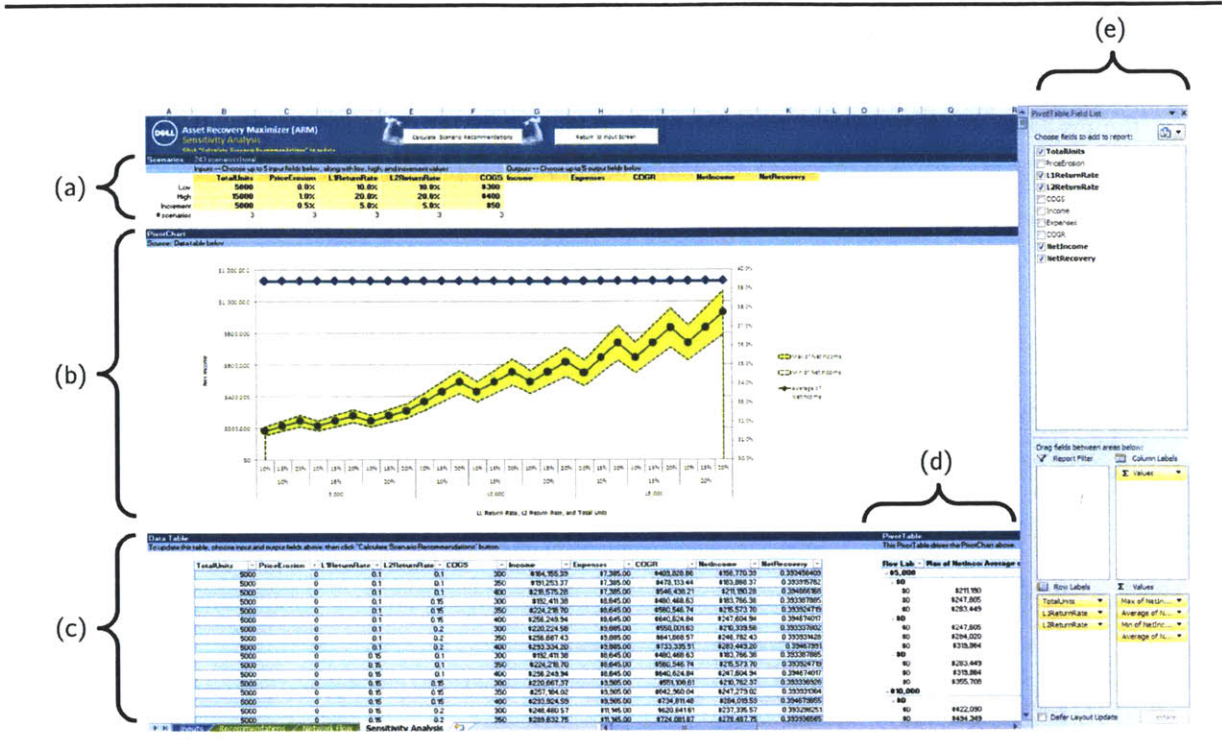


Table 39: Descriptions of the sections in the sensitivity analysis worksheet

Section	Description
Scenarios: Figure 39(a)	Provides input fields to define 1) the range of scenarios for the sensitivity analysis and 2) the calculations/recommendations to capture for each scenario.
PivotChart: Figure 39(b)	Used to graph multiple scenarios, and multiple variables from each scenario, at once—including sorting, filtering, aggregation.
DataTable: Figure 39(c)	Used as the data source to drive the PivotTable. The ARM model populates this table dynamically based on the specifications in the scenarios section.
PivotTable: Figure 39(d)	Used as the data source to drive the PivotChart.
PivotChart sidebar: Figure 39(e)	Excel sidebar to change the appearance of the PivotChart (or PivotTable).

The sensitivity analysis worksheet runs the recommendation engine multiple times, once for each sensitivity scenario. The sensitivity scenarios are all related to some baseline scenario with a subset of the input variables augmented in some way. He or she would do so by specifying the lower and upper bounds of these values to consider, and the worksheet would then, using various increments, run through all permutations of these values. For example, a user may wish to perform a sensitivity analysis on rework costs and price erosion inputs. If the range of costs is \$100/box to \$200/box with a \$25 increment, and the range of price erosion is 0%/week to 1%/week with a 0.25%/week increment, the total number of permu-

tations—or sensitivity scenarios—to investigate would be $[(200 - 100)/25 + 1] \times [(1 - 0)/0.25 + 1] = 25$ sensitivity scenarios. Across all scenarios, the other inputs of the ARM model assume baseline scenario values. As shown in Figure 40, the ARM model allows users to define sensitivity scenarios with up to five different variables involved. For each scenario, it captures up to five output fields and places the values in a data table for plotting and other types of analysis, such as linear regression.

In summary, then, the process by which a user performs a sensitivity analysis is as follows:

1. Choose a modeling scenario. We describe scenarios in Section 4.1.3
2. Specify inputs on the input screen. This provides the baseline scenario for the sensitivity analysis. We described this worksheet in Section 4.4.
3. Define the sensitivity scenarios in the scenario section. We identify this section in Figure 39(a).
4. Analyze the results using the sensitivity PivotTable and PivotChart. We identify these sections in Figure 39(d) and Figure 39(b), respectively.

Figure 40: Scenarios section of the sensitivity analysis worksheet

Scenarios 243 scenario(s) total										
Inputs – Choose up to 5 input fields below, along with low, high, and increment values. Outputs – Choose up to 5 output fields below										
	TotalUnits	PriceErosion	L1ReturnRate	L2ReturnRate	COGS	Income	Expenses	COGR	NetIncome	Recovery
Low	5000	0.0%	10.0%	10.0%	\$300					
High	15000	1.0%	20.0%	20.0%	\$400					
Increment	5000	0.5%	5.0%	5.0%	\$50					
# scenarios	3	3	3	3	3					

The performance of the ARM model varies depending on the number of sensitivity scenarios the user desires. In Table 40, we show the estimated time for the ARM model to compute various numbers of prediction scenarios. We base our “faster CPU” estimate on a runtime of 0.25 seconds per scenario, and the slower one based on a runtime of 0.40 seconds per scenario, which we measured by running multiple scenarios on various types of hardware at Dell.

Table 40: Estimated sensitivity analysis run-times

Number of scenarios	Faster CPU (e.g. dual core, > 2 GHz)	Slower CPU (e.g. single core, < 2 GHz)
10	3 seconds	4 seconds
100	25 seconds	40 seconds
1,000	4 minutes	7 minutes
10,000	$\frac{3}{4}$ hour	1 hour
100,000	7 hours	11 hours

4.7 Summary

We conclude our discussion of the ARM model with a summary of feedback from its users and ideas for future enhancements.

4.7.1 Feedback

We worked closely with ARM model working group members to refine the interface, calculations, and recommendations described in previous sections. We subsequently validated the model using several case studies. We highlight two of them in the subsequent chapters.

Along the way, our working group members demonstrated enthusiasm along the way and great excitement with the final product. Take, for example, the following feedback from two of the group's members:

- Ken Miller (ARB in EMEA): "The format is brilliant... The model provides a systematic approach to looking at things."
- Michelle Butler (GRL in DAO): "We never had this level of detail at Dell before... it's absolutely fabulous."

We believe the ARM model will help structure the decision making across regions, and also encourage the multiple reverse supply chain organizations at Dell, especially ARB and GRL, to work more closely and share best practices.

4.7.2 Future Enhancements

We imagine several straightforward enhancements that might further enhance the usefulness of the model:

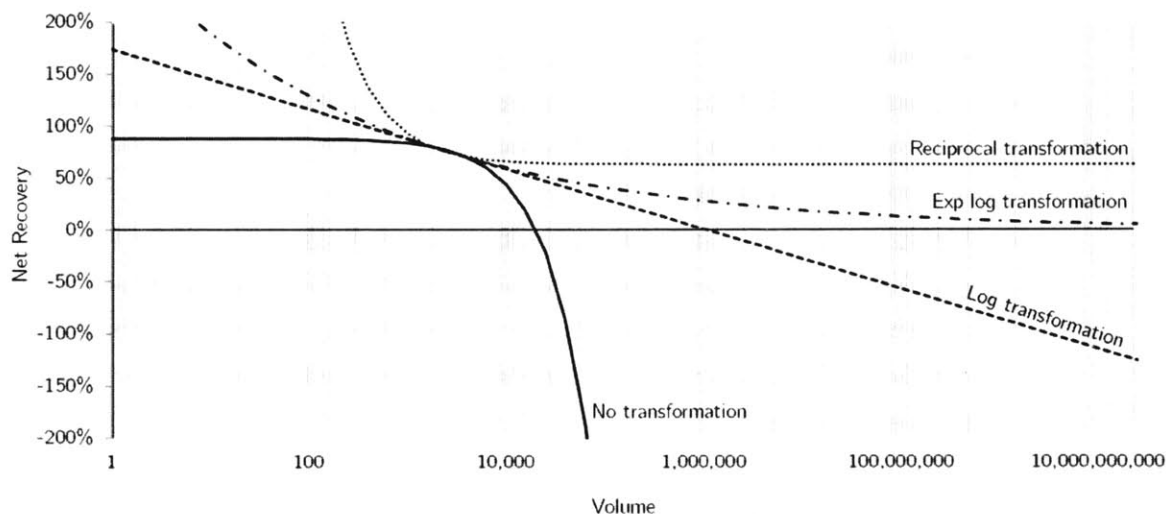
1. Convert net income estimates into net present value (NPV) terms. Today the model does not consider the time value of money. We have not yet incorporated this feature as most analyses are bounded to a year or less, and optimization results would not change even if the profit objective function were expressed in NPV terms; this is because the model already includes inventory holding costs, price erosion, and cost erosion as factors to incent inventory transformation and sales to occur as quickly as possible.
2. Extend the format to work seamlessly with the premium version of Frontline Systems' Solver. With the added capabilities, the ARM model would be able to accommodate additional rework activities, transformational activities, time periods and granularity (from quarters to perhaps weeks), and constraints.
3. Use historical data to more accurately estimate recovery channel revenue. The current version of the model assumes that the quantity sent to a channel has no impact on price; however, through our work with ARB we know that, for at least the Dell Outlet channel, greater volume correlates with lower net recovery. This happens for two reasons: one is

simple supply-and-demand economics: given a fixed demand function, a greater supply means lower prices. The other results from ARB policies; if the inventory level of a particular product exceeds a certain number of days' supply, it lowers prices to stimulate demand and lower inventory, resulting in a lower net recovery. The ARM model should understand these dynamics so it can better estimate the optimal quantity to send to a recovery channel.

Regarding the last possible enhancement, for two product families we built a linear regression model taking *available weekly sales volume* as the factor and *per-unit net recovery* as the effect. The regressions' R^2 statistics fall between 0.55 and 0.72, showing at least a modest correlation between volume and net recovery. We cannot use the resulting prediction function with the ARM model, however, due to its poor behavior at the extremes—above and below typical volumes. High volumes are especially problematic, as the function predicts nonsensical *negative* net recovery.

To correct this problem, we tried various transformations on the volume factor: logarithmic, exponential logarithmic, and reciprocal. The R^2 statistics improve slightly with these transformations, and, as shown in Figure 41, we find that the exponential logarithmic transformation provides especially nice properties at the extremes. At very low volumes, it boosts net recovery in line with the high price such scarcity would command; and at very high volumes, where the large excess would make the inventory virtually worthless, net recovery approaches an asymptote of 0%, never going negative. We believe an enhancement to the ARM model leveraging this exponential logarithmic transformation would indeed improve its net recovery estimates.

Figure 41: For net recovery prediction functions by weekly volume for an undisclosed Dell product family. The functions apply various transformations on weekly sales volume.



Chapter 5. Small Form Factor Returns Case Study

In this chapter, we present the details of a case study where we use the ARM model to help design the optimal reverse supply chain for Dell's new line of SFF products in the United States. We describe our objectives and approach, then present the findings of the case study and conclude with insights applicable beyond Dell.

5.1 Overview

Dell has recently entered the smartphone and tablet markets with a small, but growing portfolio of SFF devices, as shown in Figure 1. ARB and GRL, who are just starting to receive their first SFF product returns, are interested in the long-term impacts of SFF products on the reverse supply chain. Specifically, they seek an understanding of how the bottom-line will be impacted when they choose from various courses of action, ranging from no special handling of SFF devices to a fully customized reverse supply chain uniquely tailored to SFF devices. As we discuss in the capabilities assessment of Section 3.3.4, we believe Dell's SFF capabilities currently lie somewhere in between the two, but they are certainly less robust than its handling of longstanding laptop and desktop products.

5.1.1 Objectives

In this case study, we define four objectives:

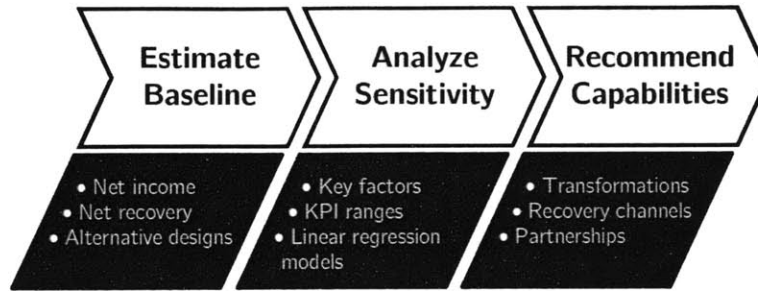
1. Identify a realistic baseline scenario. We wish to estimate of the impact of SFF returns in the reverse supply chain over the next 4 quarters.
2. Quantify the impact of uncertainty. Many important factors, such as sales volume, return rates, and price erosion are not known precisely, but are likely to fall within a certain range. We want to explore the entire range to explore the deviation from the baseline scenario.
3. Determine which factors drive net income and net recovery. Of all the factors, both known and unknown, we wish to get a sense of which drive these key metrics so as to design a robust reverse supply chain.
4. Develop SFF reverse supply chain capabilities to meet Dell's needs. Of the various transformational activities and recovery channels, we wish to identify which ones are necessary, which ones are nice, and which ones are superfluous. We know a secondary market exists for mobile phones with a solid profit margin [52], and explore whether Dell can tap that market profitably.

5.1.2 Approach

Figure 42 summarizes the three-step approach we use to meet the case study's objectives. First, using the ARM model, we estimate the necessary input parameters to arrive at our baseline scenario. By

adjusting these parameters we are able to evaluate the key metrics of alternative supply chain design. Second, again using the ARM model, we perform a sensitivity analysis by varying uncertain input factors across their range of viable options to determine the effects on these metrics. By performing linear regressions on the results, we are able to arrive at some generalizations as to the relative impacts of these factors. Third, we use this information to make recommendations about the specific actions the ARB and GRL teams should take to handle SFF returns.

Figure 42: Methodology of the small form factor case study



5.2. Baseline Scenario Estimation

To estimate our baseline scenario, we begin by defining the input parameters. We then use the ARM model to project key metrics like net income and net recovery.

5.2.1. Input Parameters

The inputs describe some characteristics of the product, as well as the “three Rs” of the reverse supply chain. We show these inputs in Table 41. As a reminder from Chapter 1, the numbers we use in this and the other case studies are *not* actual data, but still give you a sense of how we performed the analysis.) For some inputs, we use industry knowledge or literature to choose the appropriate values. Selling price erosion is one such parameter [26]. We base others, like standard cost, on internal company information. For standard cost, specifically, we take calculated the weighted-average standard cost by volume of Dell’s SFF offerings. Still, for other parameters, we forecast the values based on historical data from other areas of the company. Take return rates, for example. As a new form factor, we are uncertain what the return rates will be; however, based on the known historical return rates of laptops and desktops, for example, we can make some projections. Values for this third type of parameter are the most unreliable, of course, so those are certainly some of the ones we explore in our ensuing sensitivity analyses.

Table 41: ARM model parameters for the smartphone case study

	Parameter	Value	Range for sensitivity analysis
Product profile	Region	United States	---
	Product	All SFF devices	---
	Time horizon	Next 4 quarters (Q4FY11 – Q3FY12)	---
	Standard cost	\$300.00	\$200.00 - \$350.00
	Standard cost erosion	0%	---
	Selling price	\$400.00	---
	Selling price erosion	1%/week	0% – 1%/week
	Sales volume	300,000 units/year	100,000 – 600,000 units/year
Returns	Overall return rate	14.00% of sales volume	10.00% – 28.00%
	New condition	0.25%	---
	L1 condition	6.00%	3.00% – 13.00%
	L2 condition	6.00%	3.00% – 13.00%
	L3 or BER condition	0.75%	---
	DOA condition	1.00%	---
	E&O	0.00%	---
	Same-quarter returns	100.00%	---
Rework	L1 repair	\$50.00 cost/unit	---
	L2 repair	\$75.00 cost/unit	---
	L3 repair	Disabled	---
	ODM exchange	Exchanged for new at \$10.00 cost/unit (for shipping and handling)	---
	Teardown	Disabled	---
Recovery	Warranty stock: Refurbished or better	100% of COGS at \$10.00 cost/unit Limited to 2% of sales volume	---
	Seed stock: Refurbished or better	100% of COGS at \$10.00 cost/unit Limited to 4% of sales volume	---
	Revenue share: Refurbished or better	40% of new price at \$20.00 cost/unit	---
	Outlet sales: Refurbished or better	80% of new price at \$10.00 cost/unit	---
	Auction: BER or better	20% of COGS at \$10.00 cost/unit	---

5.2.2 Baseline Scenario Results

Through a few seemingly subtle changes with our inputs, we arrive at a menu of four different SFF reverse supply chains:

- Liquidate only: Here, the L1 and L2 repair capabilities are disabled, and most recovery channels are not available. The DOA returns may be exchanged by the ODM for new

product, which can replenish some warranty stock or seed stock, but the only available option for the de-facto non-repairable remainder is liquidation via an action process.

- Repair capabilities only: Here, repair capabilities are available, either in-house or via a 3PL, and a revenue share arrangement with a 3PL for refurbished product is also available.
- Complete repair and resale capabilities: Here, Dell has the capabilities as shown in Table 41; this includes the ability to sell refurbished product on the Dell Outlet instead of relying on a third party.
- Fully enabled with teardown: Here, we extend complete repair and resale capabilities by also adding a disassembly rework activity with a corresponding part harvesting recovery channel. That is, we allow for Dell or a 3PL to teardown SFF devices and use the parts to repair other systems.

Using the ARM model, we can directly input these four scenarios and estimate annual net income and net recovery metrics for their respective designs. We show these results in Table 42, and visualize the corresponding designs in Figure 43. The *liquidate only* option provides the lowest net income and net recovery, and the other scenarios provide progressively increasing net income and net recovery numbers. In line with intuition, a reverse supply chain with more capabilities provides a better contribution to a company's bottom line. The value the ARM model provides, then, is a quantification of the improvement. We see at least two benefits:

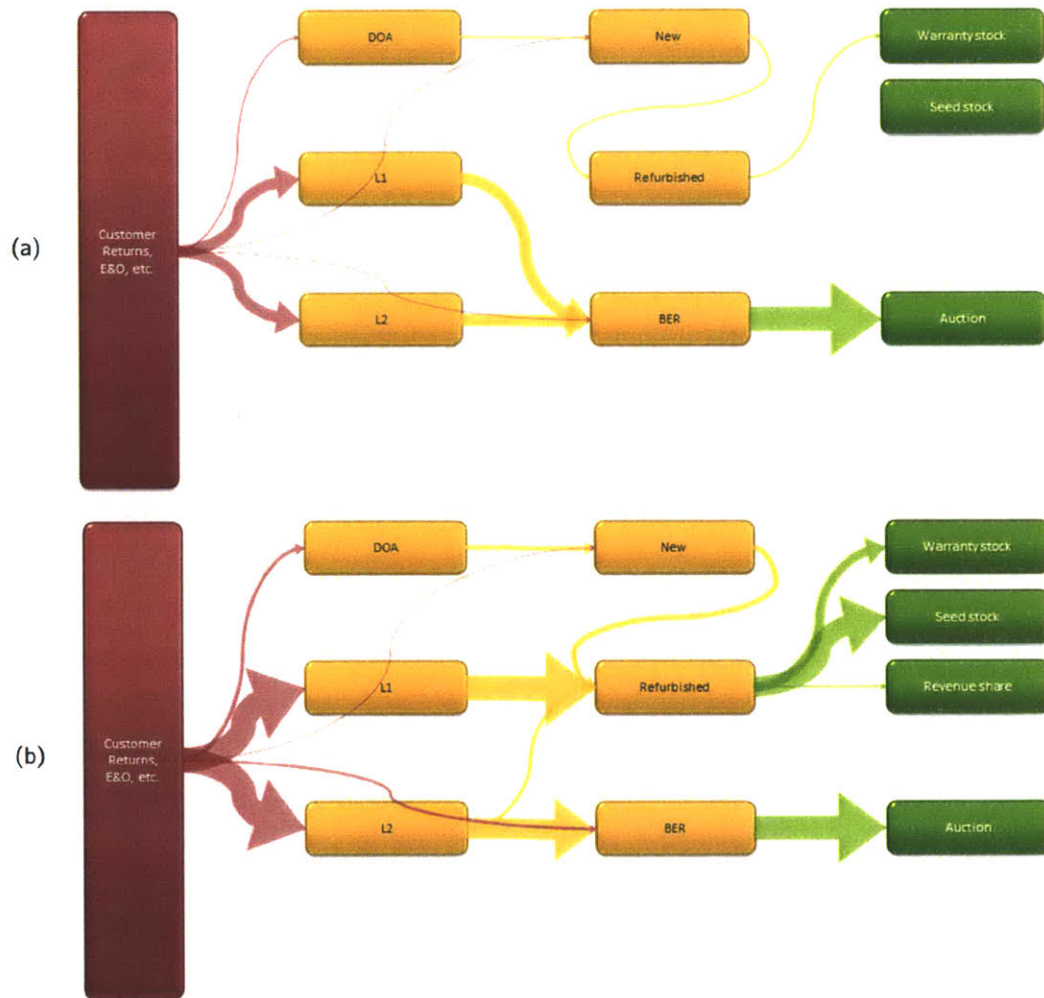
1. ARB and GRL can assess whether the current design meets the company's net recovery expectations. If not, they have other designs with better net recoveries from which to choose.
2. The net income numbers provide a basis for a cost/benefit analysis (CBA) to enhancement projects. For example, we see that the value of developing the capability of reselling SFF devices on the Dell Outlet is \$2.5M in the first year. We arrive at this number by noticing this reverse supply chain offers an estimated \$9.1M in annual net income, whereas the design without this capability offers \$6.6M, or \$2.5M less. Based on Dell's payback period and other priorities, the \$2.5M benefit may be enough for the project to win approval through the AGOP process.

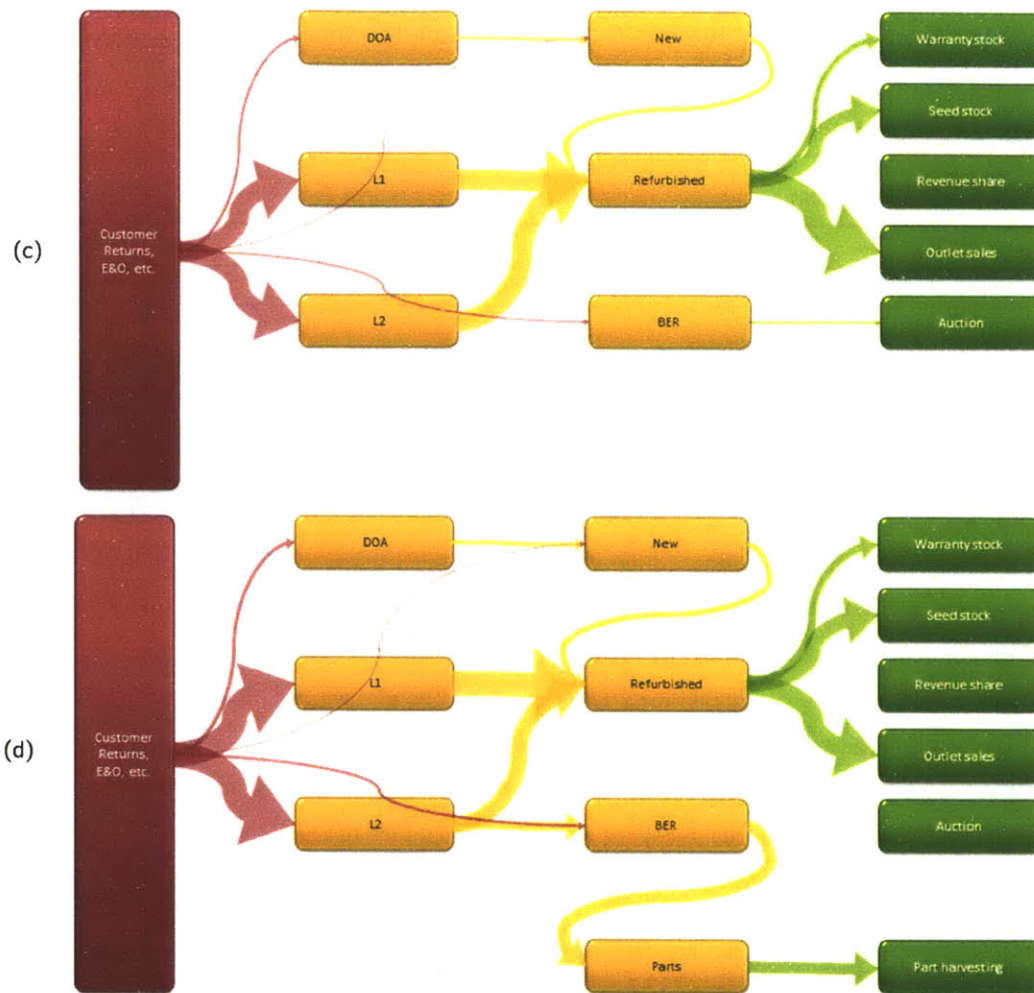
To protect confidentiality, we cannot reveal Dell's current reverse supply chain capabilities with respect to SFF devices, but we can mention that the menu of baseline scenarios yielded by this case study has helped the company assess its current capabilities and provide strategic guidance on how it should continue developing them to meet its net recovery target.

Table 42: Impact estimates of four alternative SFF reverse supply chain designs in the U.S.

Design	Net income	Net recovery	LCM net recovery
(a) Liquidate only	\$3.2M	28%	29%
(b) Repair capabilities only	\$6.6M	58%	59%
(c) Complete repair and resale capabilities	\$9.1M	77%	79%
(d) Fully enabled with teardown	\$9.8M	82%	83%

Figure 43: Cumulative (4-quarter) network flow diagrams for four alternative SFF reverse supply chain designs: (a) liquidate only, (b) repair capabilities only, (c) complete repair and resale capabilities, and (d) fully enabled with teardown





5.3. Sensitivity Analysis

With respect to the baseline scenarios, we acknowledge that the net income and net recovery projections are based on estimates. Some of the parameters are, at the current time, hard to forecast accurately. We proceed with a sensitivity analysis on these parameters to get a better sense of how robust the designs are to erroneous assumptions and changing market condition. For the purpose of clarity and brevity in our analysis, we focus on one of the four baseline scenarios: complete repair and resale capabilities as shown in Figure 43,c.

5.3.1. ARM Model Results

We use the sensitivity analysis worksheet of the ARM model to optimize 768 modeling scenarios, based on the sensitivity ranges shown in Table 43. The other inputs are as shown in

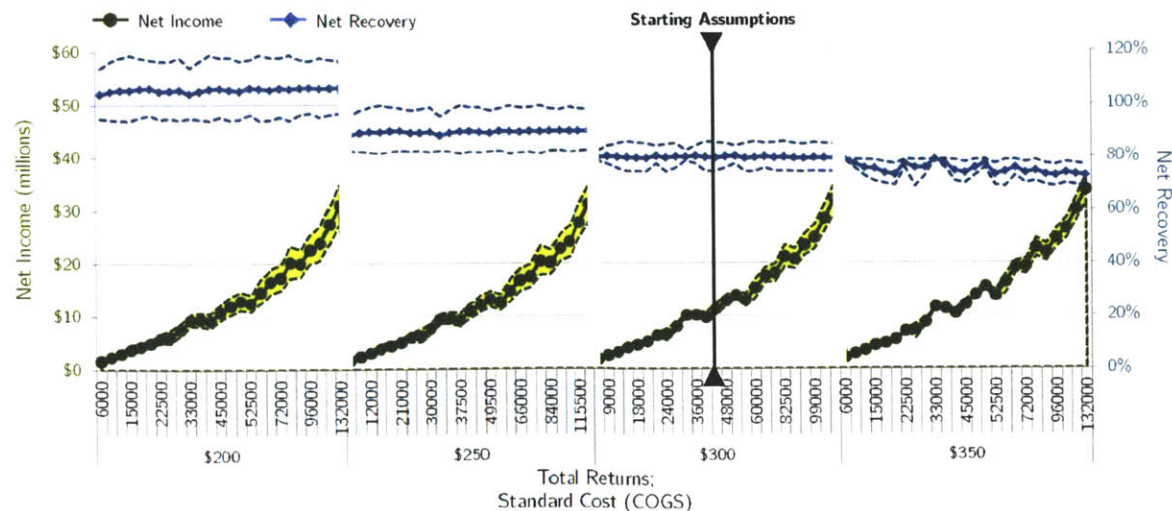
Table 41. To interpret the results, we manipulate the resulting Excel PivotChart in various ways. When we plot the total number of returns and standard cost against net income and net recovery, we produce the graph shown in Figure 44.

Table 43: Input fields and value ranges used to define the modeling scenarios for the SFF returns sensitivity analysis

Parameter	Low	High	Increment	Number of scenarios
Total units sold	100,000	600,000	150,000	4
Price erosion	0.00%/week	1.00%/week	0.50%	3
L1 return rate	3.0%	12.0%	3.0%	4
L2 return rate	3.0%	12.0%	3.0%	4
Standard cost	\$200.00	\$350.00	\$50.00	4
Total number of scenarios: 768				

In addition to the two series, which we format as solid lines, we also show a band represented by the range of values from the minimum to maximum. The width of these bands are controlled by the hidden factors not plotted in the graph: L1 return rate, L2 return rate, sales volume, and sales price erosion. By plotting various factors and examining the size of the bands, we can deduce which factors yield the highest rate of variability. In Figure 44, for example, the net income bands are all very tight and virtually identically across standard costs, indicating that the total return rate accounts for most of the variability in this metric; whereas with net recovery the step appearance of the lines show the critical role standard cost plays in the number while the total return rate plays no role at all.

Figure 44: Sensitivity analysis plot for key factors in the SFF case study, showing positive net income across all scenarios with 67-119% net recovery

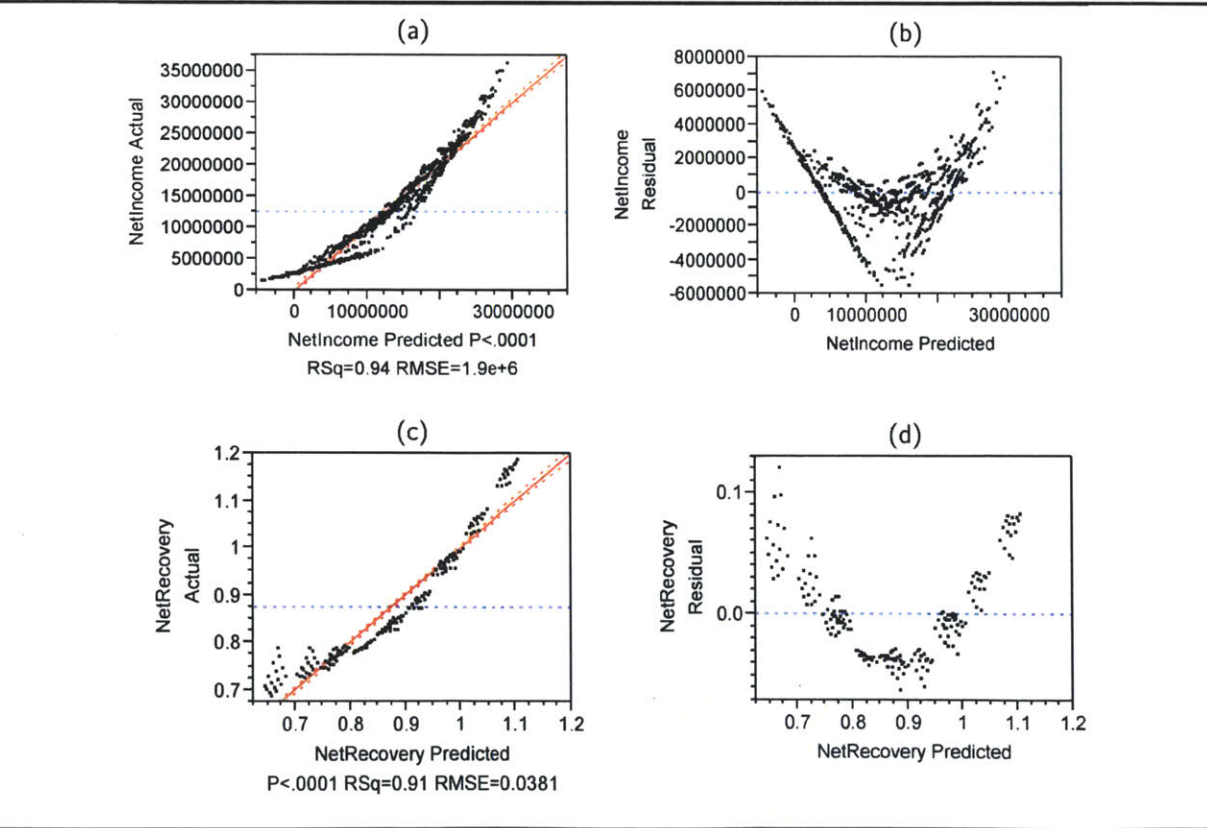


In summary, the graph indicates that despite the broad range of inputs tested, the supply chain design holds up well. It provides a net recovery of 67% or better, averaging 87% across the 768 scenarios. Further, all of the scenarios considered yield positive net income, ranging from \$1.5M to \$36.2M, averaging \$12.5M.

5.3.2. Linear Regression

For a more thorough analysis of the factors' respective influences on net income and net recovery, we export the sensitivity analysis results from the ARM model into a statistical package. We use the parameters from Table 43 as factors and define net income and net recovery as effects. The results are shown in Figure 45. Both effects appear with non-random residual plots, indicating we are missing important factors or interactions, however the fit is still good: we see adjusted R^2 statistics of 0.934 and 0.914 for net income and net recovery, respectively. With the net income regression, all factors are significant at $p < 0.0001$. With the net recovery regression, all factors are significant except total units sold.

Figure 45: Residual plots of SFF sensitivity analysis linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual








Using coefficients from the predicted linear expressions, we build Table 44 to show the incremental effects of individual factors. The table confirms our suspicions from the earlier analysis while allowing us to add some precision. With respect to net income, we see total units sold—a proxy for total number

of returns—as the key factor, driving almost \$2M in additional net income with each 50,000 additional sales. Secondly, the L1 and L2 return rates have important effects, driving close to three quarters of a million dollars in additional net income for every percentage increase in returns. This is welcome news, in a sense. Even if product returns are higher than expected, Dell can at least recover a good portion of the assets' value.

With respect to net recovery, other factors prove more relevant. Specifically, for every \$15 increase in standard cost, net recovery decreases by 3.1%; and, for every 0.1% increase in price erosion, net recovery decreases by 1.2%. These declines make sense, as the corresponding increases erode sales price-based revenue in the numerator of the net recovery calculation, while the denominator remains constant. Surprisingly, we see that increases in the L1 return rate boost net recovery, whereas increases in the L2 return rate diminish it. Upon tracing the flow of product in the ARM model, we see why: ARM recommends refurbishing all of L1 and L2 product for seed stock replenishment or outlet sales. L2 repair, being more expensive than L1 repair, yields net recoveries slightly below average, whereas the opposite is true for L1 repair.

Table 44: Sensitivity analysis of SFF returns based on key factors

Factor	Range tested	▲=Starting assumption	Incremental net income impact	Incremental net recovery impact
Total units sold	100K  600K		+\$1.92M per 50K	No impact
L1 return rate	3%  13%		+\$0.76M per 1%	+0.2% per 1%
L2 return rate	3%  13%		+\$0.68M per 1%	-0.3% per 1%
Standard cost (COGS)	\$200  \$350		+\$0.15M per \$15	-3.1% per \$15
Price erosion per week	0.0%  1.0%		-\$0.21M per 0.1%	-1.2% per 0.1%

The sensitivity analysis allows us to make better-informed decisions on reverse supply chain commitments. If we are especially interested in a supply chain that provides a baseline level of net income, we will want to focus our attention on the return volume. If, instead, we are more interested in ensuring a net recovery target, we must focus on refining our estimate for standard cost, or even break down our SFF analysis into constituent parts for specific SFF devices at different standard costs to ensure we have a robust design for the various devices.

5.3.3. Second-Pass Linear Regression

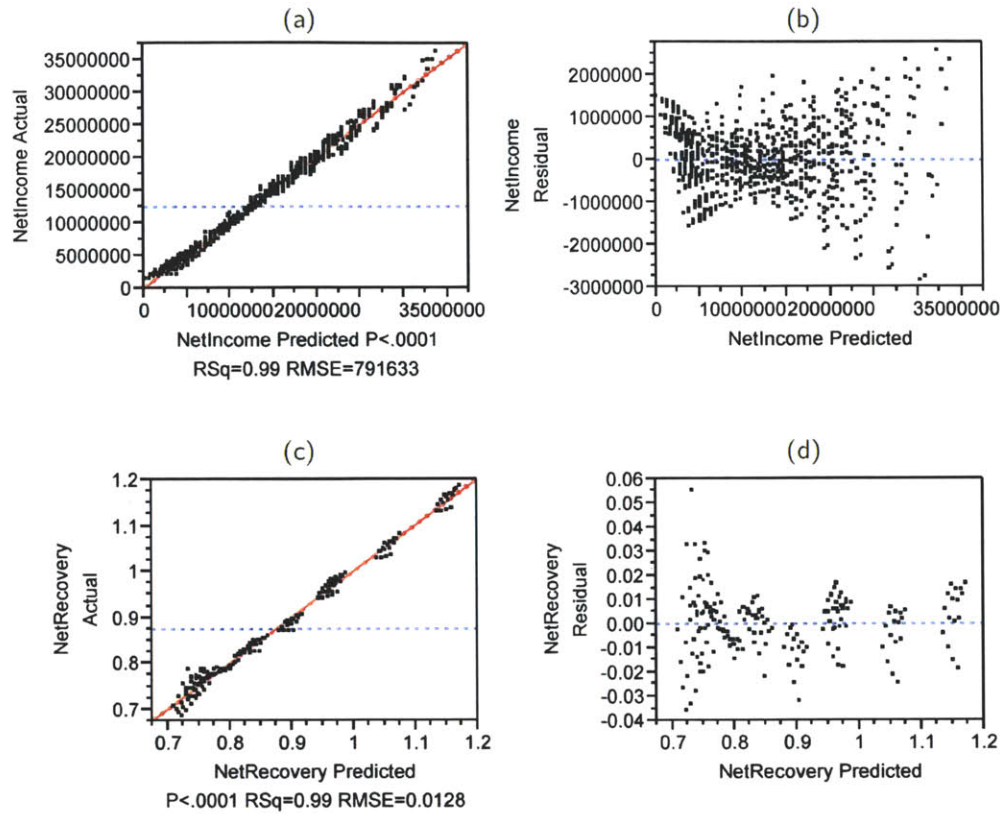
The plots in Figure 45 are somewhat problematic: the predicted vs. actuals and predicted vs. residuals plots show a curvature in the residuals, instead of appearing randomly, meaning we are not accounting for some factor(s) in our model. We take another pass at our linear regression analysis to find a

better fit. Our results are in Table 45 and the corresponding residual plots in Figure 46. We find that these refined models do not provide results significantly different enough from those in Table 44 to sway our analysis. This may not always be the case, however. Depending on the reverse supply chain design, the simpler linear regression may not be sufficient, specifically when multiple nodes in the network feature binding capacity constraints from which the ARM recommendation engine will likely yield jagged, non-linear net income and net recovery predictions.

Table 45: SFF sensitivity analysis second-pass linear regression results

	Net income	Net recovery
Adjusted R ²	0.989	0.990
Additional factors	<ul style="list-style-type: none"> • Total returns (significant with $p < 0.0001$) 	<ul style="list-style-type: none"> • COGS x COGS • COGS x price erosion (both significant with $p < 0.0001$)
Prediction expression	-1739114.9088397 $-211781450.28515 \times \{\text{Price erosion}\}$ $+10176.0247360417 \times \{\text{Standard cost}\}$ $+5.23824327415614 \times \{\text{Total units}\}$ $+220.686092048679 \times \{\text{Total returns}\}$	1.4693765882655 $-11.768641626172 \times \{\text{Price erosion}\}$ $+0.1521066768402 \times \{\text{L1 return rate}\}$ $-0.2560952976736 \times \{\text{L2 return rate}\}$ $-0.0020456836723 \times \{\text{Standard cost}\}$ $+0.09855988517812 \times \{\text{Standard cost}\}$ $\quad \times \{\text{Price erosion} - 0.005\}$ $+0.00001113415405 \times \{\text{Standard cost} - 275\}^2$

Figure 46: Residual plots of SFF sensitivity analysis second-pass linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual



5.4. Capability Recommendations

Based on our analysis, and in conjunction with our market research and conversations with 3PL candidates, we used our ARM analysis to advise Dell on their SFF reverse supply chain design. Broadly, we note the importance of a well-enabled reverse supply chain with repair and resale capabilities (Figure 43(c)). To build this capability, we recommend a variety of partnerships. We see teardown capabilities (Figure 43(d)) as an ideal state, though its benefits may not justify its enablement costs. Armed with the ARM model, we leave that CBA and additional analysis to the GRL and ARB teams.

5.5. Generalizable Insights

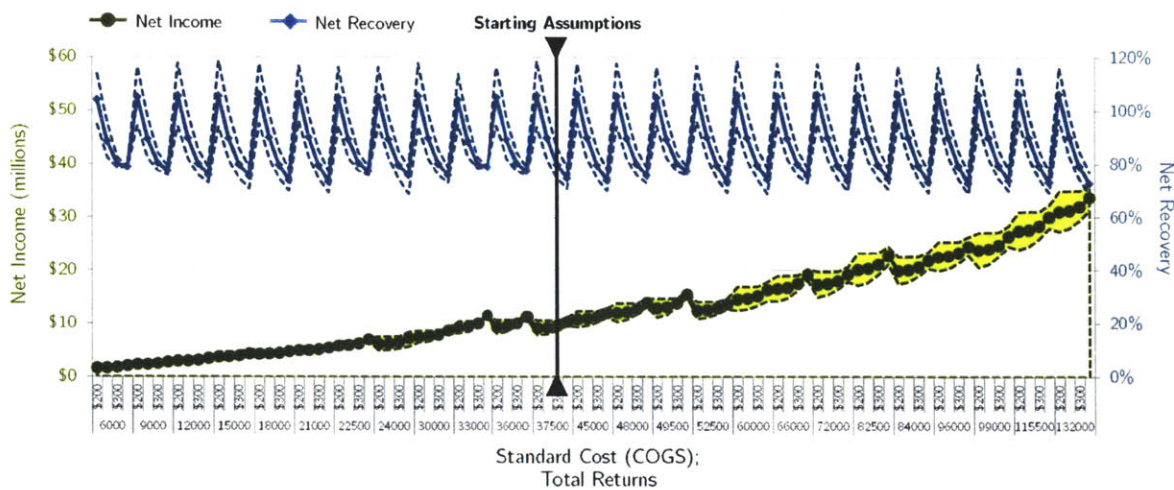
In this case study, we see interesting interplay between the net income and net recovery metrics. There are two generalizable conclusions:

1. The factors that affect net income and net recovery may be different. As we show in the sensitivity analysis of Figure 44 and the linear regression analysis of Table 44, net income can remain constant while net recovery increases, and vice versa. A firm should focus on

the factors that influence the metric it cares about most, while being mindful of the factors that influence the others.

2. Net income and net recovery may move in opposite directions. In plotting the modeling scenario results in the ARM model, we observed some striking ways in which net income increases while net recovery decreases (and vice versa), such as in Figure 47. As standard cost and total returns increase, so does net income, generally speaking. Net recovery, however, decreases as standard cost increases. This has an important management implication: if you are managing an organization with net recovery a marquee metric, use it with caution. A boost in net recovery does not necessarily correspond with an improvement to the bottom line; in fact, it could mean the opposite! Of course, the opposite is true also; a faltering net recovery need not reflect declining performance.

Figure 47: Plot of showing net income (green line) and net recovery (blue line) moving in opposite directions. Regardless of return volume, as COGS increases net income increases while net recovery decreases.



Additionally, we believe any firm can leverage the methodology we propose in Figure 42 to quantify an opportunity and evaluate various supply chain designs. The approach works particularly well for a new product or form factor, where the firm may wish to estimate expected net income and net recovery metrics. Net income, if sufficiently large, can motivate a group to take action and provide the justification for prioritizing related initiatives. Net recovery, if below expectations, can also be a motivating force for action. In general, the ARM model provides a straightforward way to estimate these numbers. Even in the face of uncertainty, estimates are still possible by using its sensitivity analysis spreadsheet. If robust estimates are required, the sensitivity analysis may be exported to a statistical package where it can be used as the basis for a linear regression. With a linear regression in hand, one may assign ranges or distributions to the factors and plot a distribution of the predicted effects.

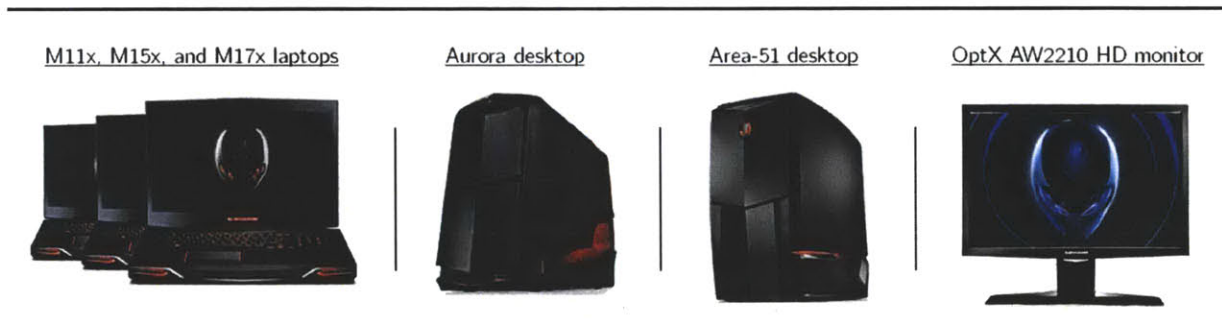
Chapter 6. Alienware Repair Case Study

In this chapter, we describe a case study we performed with the ARB and GRL groups in EMEA regarding enabling repair capabilities for Alienware-branded systems. In contrast to the SFF case study, here we explore a longstanding product line with a well-established reverse supply chain. That is, instead of looking at determining an optimal initial supply chain design, we are looking to extend its capabilities by incorporating repair. We begin with an overview of the study where we state our objectives and approach. We then proceed by walking through the approach, from calibration to impact analysis to sensitivity analysis. Finally, we pull together relevant insights that extend beyond this particular case study.

6.1. Overview

Alienware is a line of high-performance gaming desktops, laptops, and peripherals with distinctive sci-fi markings and illumination, including the sampling we provide in Figure 48. The brand originated from a company with the same name in 1996, which Dell acquired ten years later, in 2006. In the U.S. reverse supply chain, Dell's handling of Alienware systems has mirrored that of other systems for years. That is, Dell employs the same process as it uses with Dell-branded hardware, including EMR repair and teardown transformation as necessary. In EMEA, on the other hand, the reverse supply chain has historically not included any repair capability.

Figure 48: Sampling from the Alienware product line



This case study endeavors to determine whether EMEA should begin repairing Alienware systems. We examine Alienware systems specifically as opposed to *all* system returns in EMEA at the direction of the EMEA-based ARB and GRL members of our working group. Interested in launching a pilot program to repair product returns, they selected Alienware systems as a manageable subpopulation with comparatively high margins. If the pilot did not prove financially viable with Alienware systems, it would likely not work with any other kind of system. Though this case study, we discovered incorporating repair capabilities would add millions of dollars to Dell's bottom line.

6.1.1. Objectives

In this case study, we define three objectives:

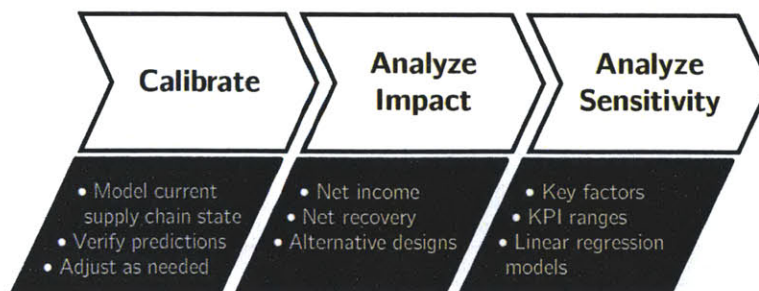
1. Validate the ARM model with a baseline scenario. We do this to calibrate the model and build its credibility within the ARB and GRL groups.
2. Determine the impact of adding repair capabilities. We hypothesize that the new capabilities will provide benefits to Dell in terms of an increase in both net income and net recovery, however we wish to quantify these benefits in terms of net income and net recovery.
3. Assess the volatility of the benefits in the face of uncertainty. We must identify factors with uncertainty, such as the demand for refurbished Alienware systems in EMEA, and see the degree to which this impacts net income and net recovery.

6.1.2. Approach

The three objectives translate directly into our three-step approach for this case study: calibrate the model, perform an impact analysis, and perform a sensitivity analysis. We summarize the approach in Figure 49. The first step, calibration, was not possible in the SFF returns case study of Chapter 5 since Dell had no historical data on handling SFF returns. Here, in EMEA, there is data on the current performance of the reverse supply chain with respect to Alienware systems. We want to start by calibrating the model to validate we have the parameters correct.

Subsequently, we adjust a few of the parameters to add repair capabilities, and measure the change in net income and net recovery. Finally, we perform a sensitivity analysis using the ARM model and linear regression to set some likely boundaries on how good, or how bad, the new capability may be for Dell. We discuss the findings from these three steps in the next sections.

Figure 49: Methodology of the Alienware repair case study



6.2. Calibration

To calibrate the ARM model, we supply input parameters representing the current state, ask it to generate recommendations, and then compare the results to actual values. In collaboration with ARB and GRL group members, we make adjustments as needed.

6.2.1 Input Parameters

We show the input parameters in Table 46. As stated previously, the values we present are not actual data, which is confidential to Dell, but rather fictitious data that promotes similar analysis and yields similar conclusions. For product profile parameters, we use a volume-based weighted average for the price and standard cost of Alienware systems. For return rates, we use historical information. Incidentally, return rates in EMEA are dramatically lower than in the U.S.; we posit a number of reasons for this, including cultural differences. For rework options, we specify no options in the calibration base case; and for recovery channels, we represent the third-party broker channels Dell currently uses in EMEA. The outlet sales channel is not present in the calibration base case either.

Table 46: ARM model parameters for the Alienware repair case study

	Parameter	Value	Range for sensitivity analysis
Product profile	Region	EMEA	---
	Product	All Alienware systems	---
	Time horizon	Next 4 quarters (Q4FY11 – Q3FY12)	---
	Standard cost	\$1,200.00	---
	Standard cost erosion	0%	---
	Selling price	\$1,500.00	---
	Selling price erosion	1.00%/week	0.00% – 1.00%/week
	Sales volume	300,000 units/year	---
Returns	Overall return rate	3.50% of sales volume	---
	New condition	0.25%	---
	L1 condition	1.25%	---
	L2 condition	1.00%	---
	L3 condition	0.50%	---
	BER condition	0.50%	---
	DOA condition	0.00%	---
	Same-quarter returns	100.00%	---
Rework	L1 repair	\$150.00 cost/unit (disabled for calibration)	\$100.00 – \$200.00 cost/unit
	L2 repair	Disabled	---
	L3 repair	Disabled	---
	ODM exchange	Disabled	---
	Teardown	Disabled	---
Recovery	Broker/sealed: New only	70% of COGS at \$40.00 cost/unit	---
	Broker/standard: BER or better	30% of COGS at \$40.00 cost/unit	---
	Outlet sales:	70% of new price at \$40.00 cost/unit	40% – 100% of new price
	Refurbished or better	Limited to 500/quarter (disabled for calibration)	250 – 850/quarter

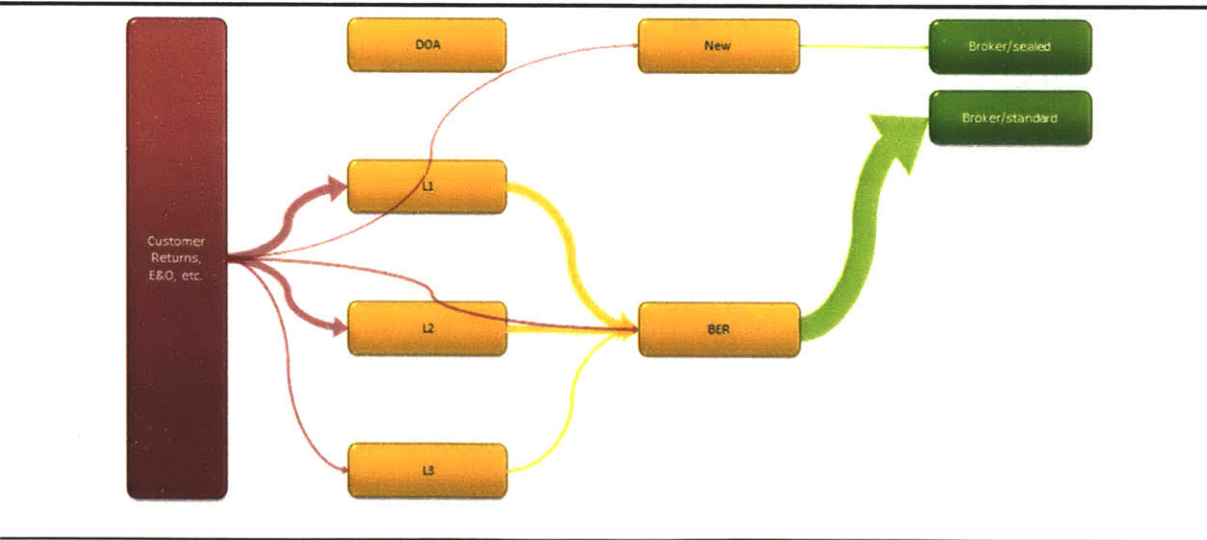
In Table 46 we also identify which parameters are uncertain by specifying a range of values we wish to explore during our sensitivity analysis.

6.2.2. Baseline Scenario Results

Given the model parameters for the baseline scenario, the ARM model predicts \$3.7M in annual net income and a 32% net recovery rate. It also generates the network flow diagram in Figure 50. As expected, it shows an absence of rework activities and limited recovery channels, with the vast majority of product returns sent to the broker/standard channel.

As it turns out, the ARM model's output closely matches ARB and GRL's metrics of the current state, so we are able to declare calibration complete and consider the model verified. Had the estimates been inaccurate, we would have reviewed the parameters and re-run the model with some adjustments, such as with a slightly different selling price. Had the results still not been satisfactory, we would have revisited some of the assumptions in the underlying model.

Figure 50: Cumulative (4-quarter) network flow diagram representing the current state reverse supply chain for Alienware repair in EMEA (i.e. no repair option)



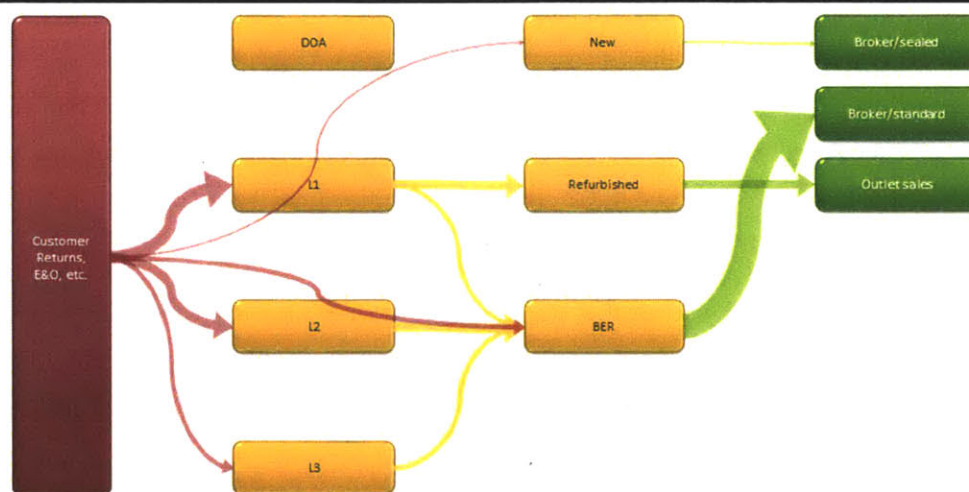
6.3. Impact Analysis

With the ARM model validated, we now adjust the inputs to incorporate the repair option and new Dell Outlet recovery channel or refurbished product. The repair option Dell is considering is a low-touch level one (L1) repair only, for cosmetic defects and simple hardware swaps; consequently, L2 and L3 returns cannot be repaired even with L1 repair capability.

We show the resulting network flow diagram in Figure 51. The ARM model opts to use the repair option, which is represented by the flow of product returned in L1 condition to the *Refurbished* node.

This proportion of product is ultimately routed to the Dell Outlet. Not all L1 product is routed to the *Refurbished* node, however, due to a capacity constraint we set at the Dell Outlet. To keep the scenario conservative, ARB did not want to assume it would be able to sell a large volume of refurbished Alienware systems in the Outlet, so we limited its capacity to 500/quarter. The ARM model, noting this constraint, elected to repair no more than the amount needed to meet this demand; for the remainder, it rightly observed that spending money on repairs would not be rewarded by a higher price in the broker/standard channel.

Figure 51: Cumulative (4-quarter) network flow diagram with the Alienware repair capability enabled in EMEA



In Table 47, we summarize the differences between the current reverse supply chain and one with the new repair capabilities. The new design boosts net income by an estimated 22% to \$4.5M, and boosts net recovery by eight percentage points, to 41%. The network flow diagrams of the two designs do not look much different, but the additional repair capabilities certainly offer an appealing benefit to Dell.

Table 47: Comparison of two alternative supply chain designs for Alienware returns in EMEA: (a) no repair, and (b) repair-with-resale capabilities

Design	Net income	Net recovery	LCM net recovery
(a) No repair	\$3.7M	32%	33%
(b) Repair and outlet capabilities	\$4.5M	40%	41%

These estimates do not consider any fixed costs associated with repair capabilities, such as equipment purchases or other SG&A. While the ARM model has the ability to account for one-time set-up costs, ARB and GRL did not wish to consider any given its arrangement with the 3PL Flextronics to manage its reverse supply chain in EMEA. Under the arrangement, these costs are amortized over the variable ABC of repair, which we assess at \$150 per box.

6.4. Sensitivity Analysis

So far, we have made assumptions regarding at a couple uncertain parameters: the actual average cost of repairs, and demand constraints on Alienware systems in an EMEA outlet. In this section, we explore the impact of variability with these and other factors.

6.4.1 ARM Model Results

We use the ARM model's sensitivity analysis capabilities to generate optimal supply chain designs for 630 sensitivity scenarios. We explore a range of values for a number of parameters—refurbished price, price erosion, repair cost per box, and refurbished sales capacity—as shown in Table 48. As in the SFF returns case study, we plot the data in a PivotChart to help us analyze the results.

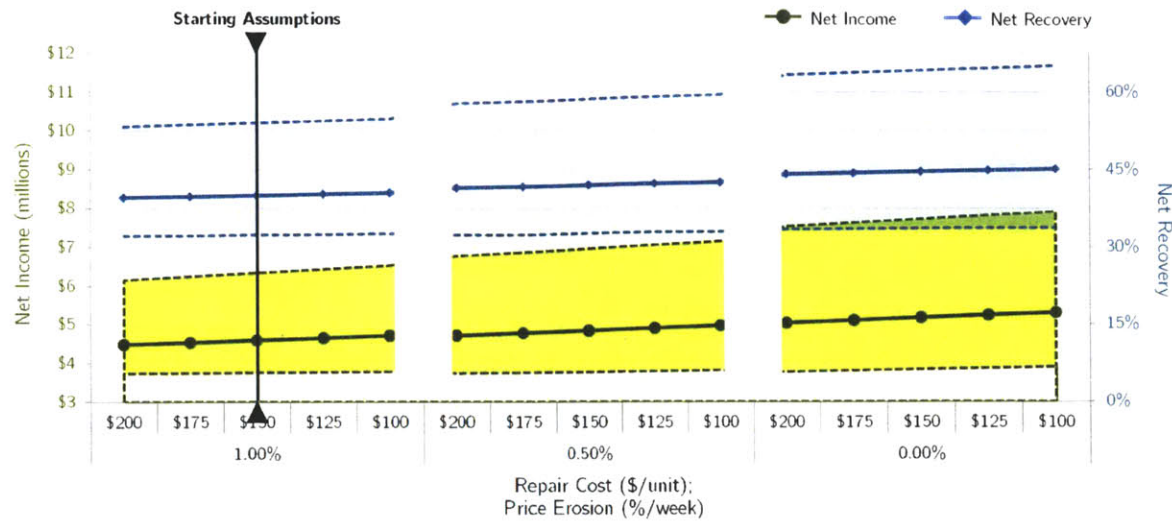
Table 48: Input fields and value ranges used to define the modeling scenarios for the Alienware repair sensitivity analysis

Parameter	Low	High	Increment	Number of scenarios
Refurbished price	40% of new price	100% of new price	10%	7
Price erosion	0.00%/week	1.00%/week	0.50%	3
Repair cost per box	\$100/unit	\$200/unit	\$25	5
Refurbished sales capacity	250 units/quarter	850 units/quarter	120	6
Total number of sensitivity scenarios:				630

In the first, Figure 52, we examine the effects of repair cost and price erosion on net income and net recovery. The green bands convey the range of possible values at each net income data point based on the parameters not pictured: refurbished price and refurbished sales capacity. The blue bands convey the range of possible values for net recovery. What we see in this graph is that, unexpectedly, repair cost is a negligible factor. As the repair cost halves from \$200/unit to \$100/unit, estimated net income increases by only 5% and net recovery by 2%, despite the fact that repaired systems constitute a quarter of total return volume. The large price differential between selling through a broker and selling refurbished through the Dell outlet dwarfs the repair costs. This is good news; even if the average repair cost turns out to be higher than the \$150/unit estimate, the impact will not be great. The converse is also true: driving efficiencies in the repair process will not have a big impact on net income, at least with the anticipated low volume of 5.5 repairs/day.

Price erosion appears to be a bigger factor, but not by a large amount. If there were no price erosion instead of the assumed 1.0%/week price erosion, net income and net recovery would be about 12% larger. This demonstrates the importance of incorporating price erosion into the ARM model; without multi-period support or a price erosion mechanism, the model's estimates could easily be off by 10% or more.

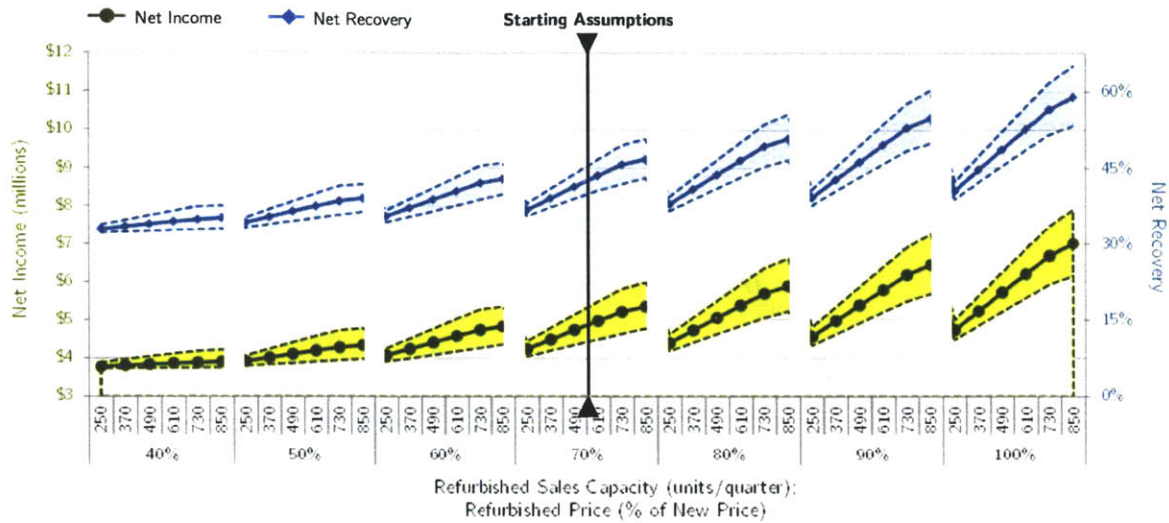
Figure 52: Sensitivity analysis plot for repair cost and price erosion in the Alienware repair case study



In Figure 53, we plot the other two factors, refurbished sales capacity and refurbished price. Here we see a much broader range of responses and much tighter bands between the minimum and maximum values. This tells us that these two factors explain most of the variation in terms of net income and net recovery. Also noteworthy, and unlike what we saw in the SFF returns case study, is how the two metrics move in the same direction: as net income increases, so does net recovery. We also see the increased leverage of refurbished sales capacity as the refurbished price increases. At the low end, where the refurbished price is 40% of the new price, doubling outlet sales capacity increases net income by 2%. Doing so when the refurbished price is a more realistic 70% of the new price, however, increases net income by 16%.

Between the two plots, we observe that under the worst-case assumptions of high repair costs, high price erosion, low resale value, and low refurbished price, adding repair capabilities to the supply chain does not improve net recovery or net income from the current state. It remains at \$3.7M/year with a 32% net recovery. Since ARB and GRL would be hard-pressed to justify any management overhead of a new repair capability that added no value, we would not want to enable repair capabilities if this worst-case scenario were at all likely. If, on the other hand, ARB and GRL are confident they can command a refurbished price of at least 60% of new, however, even if all the other factors turned out to retain their worst-case status, the repair capability is still desirable. Net income and net recovery would increase by 5%. Based on historical sales in the Outlet, we consider a 60%-of-new sales price to be very conservative and a 70%-of-new (or greater) price to be more realistic. Consequently, we recommend moving adding repair capability to Alienware systems in EMEA's reverse supply chain.

Figure 53: Sensitivity analysis plot for refurbished sales capacity and refurbished price in the Alienware repair case study



6.4.2. Linear Regression

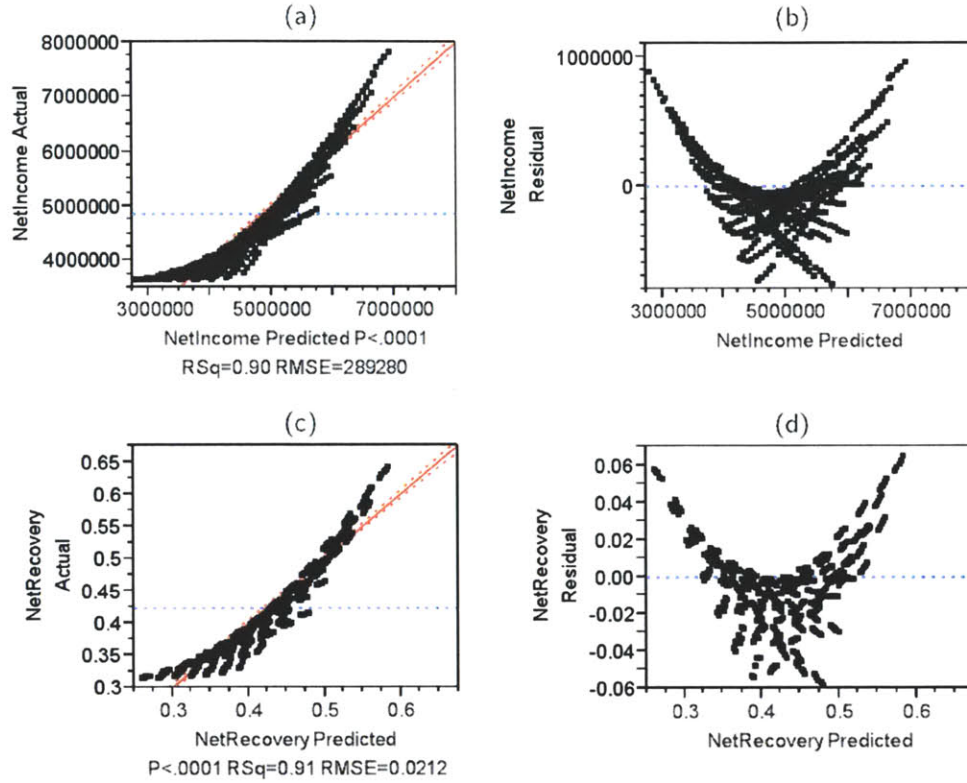
We export the results from the ARM model's sensitivity analysis into a statistical package for further analysis. We build two linear regression models, one for net income and another for net recovery. In both cases, the four parameters we explored in the sensitivity analysis are factors in each of the models. Using the coefficients from the prediction expressions, we present in Table 49 the incremental effects of adjusting any parameter (factor) by 10% of its range tested. The results confirm what we deduced above; namely, that refurbished price is the key factor influencing both net income and net recovery, with refurbished sales capacity an important secondary factor. Each 6% boost in refurbished price adds about \$210K to net income, whereas the corresponding 60 unit increase in sales capacity adds about half that. Much less impactful are the price erosion and repair cost factors, which contribute negligibly to either net income or net recovery.

Table 49: Sensitivity analysis of Alienware repair based on key factors

Factor	Range tested	↕=Starting assumption	Incremental net income impact	Incremental net recovery impact
Refurbished price (% of new price)	40% 100%		+\$210K per 6%	+1.6% per 6%
Price erosion per week	0.0% 1.0%		-\$57K per 0.1%	-0.5% per 0.1%
Repair cost per box	\$100 \$200		-\$25K per \$10	-0.1% per \$10
Refurbished sales capacity (units/quarter)	250 850		+\$118K per 60	+1.0% per 60

In Figure 54 we show the residual plots of the linear regressions. While the adjusted R^2 values are fairly high in both models, at 0.895 for net income and 0.911 for net recovery, the sloping of the residuals speak to interactions we should consider. We do this next.

Figure 54: Residual plots of Alienware repair analysis linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual



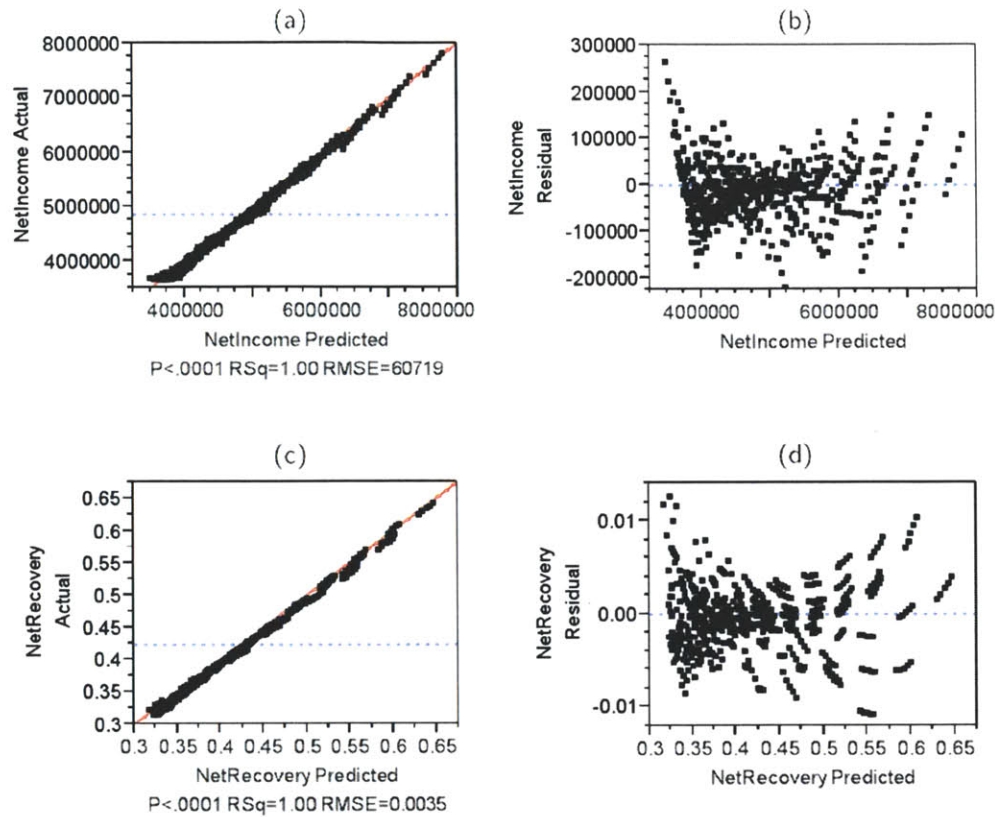
6.4.3. Second-Pass Linear Regression

We refine our linear regressions by adding interaction factors, as shown in Table 50. The key interactions for both models are refurbished capacity with various other factors and refurbished price with other factors. We noticed one difference between the two models: *refurbished price* \times *refurbished price* is significant for net income, whereas *refurbished capacity* \times *repair cost* is significant for net recovery. The resulting adjusted R^2 values are very high, and the residual plots in Figure 55 look much more random.

Table 50: Alienware repair analysis second-pass linear regression results

	Net income	Net recovery
Adjusted R ²	0.995	0.998
Additional factors	<ul style="list-style-type: none"> • Refurb capacity × price erosion • Refurb capacity × refurb capacity • Refurb capacity × refurb price • Refurb price × refurb price • Refurb price × price erosion • Price erosion × price erosion <i>(all significant with p < 0.0001)</i>	<ul style="list-style-type: none"> • Refurb capacity × price erosion • Refurb capacity × refurb capacity • Refurb capacity × refurb price • Refurb capacity × repair cost • Refurb price × price erosion • Price erosion × price erosion <i>(all significant with p < 0.0001)</i>
Prediction expression	1966528.46693253 + 3505038.45361112 × {Refurb price} + 1962.1155022222 × {Refurb capacity} − 57292085.561905 × {Price erosion} − 2477.2178260317 × {Repair cost} + 552340.924206345 × {Refurb price − 0.7} ² + 6098.54517148526 × {Refurb price − 0.7} × {Refurb capacity − 550} − 0.663964712774 × {Refurb capacity − 550} ² − 117183515.59524 × {Refurb price − 0.7} × {Price erosion − 0.005} − 96146.600108844 × {Refurb capacity − 550} × {Price erosion − 0.005} + 1594577798.09522 × {Price erosion − 0.005} ²	0.17534505202046 + 0.27100109601587 × {Refurb price} + 0.00017416663875 × {Refurb capacity} − 4.5967052495238 × {Price erosion} − 0.0000968337704 × {Repair cost} + 0.00045948703406 × {Refurb price − 0.7} × {Refurb capacity − 550} − 0.000000073593751653 × {Refurb capacity − 550} ² − 6.6423443404761 × {Refurb price − 0.7} × {Price erosion − 0.005} − 0.0074880333973 × {Refurb capacity − 550} × {Price erosion − 0.005} + 107.566559619045 × {Price erosion − 0.005} ² − 0.00000018570765338 × {Refurb capacity − 550} × {Repair cost − 150}

Figure 55: Residual plots of Alienware repair analysis second-pass linear regression: net income effect as (a) predicted vs. actual and (b) predicted vs. residual; net recovery effect as (c) predicted vs. actual and (d) predicted vs. residual



6.5. Generalizable Insights

Based on the recommendations of our case study, which showed benefits to adding repair capabilities so long as the refurbished price would be at least 60% of the new price, ARB and GRL began repairing Alienware systems in EMEA starting in the last quarter of 2010. As of this writing, we do not have enough data to assess the accuracy of the ARM model's predictions, but the EMEA groups feel confident in the program's success and hope to expand their repair capabilities beyond Alienware systems in 2011.

One of the interesting insights from this case study is the relative importance of various factors. Intuitively, repair cost would seem an important factor, however the analysis revealed that it can be rather insignificant; refurbished price and resale demand capacity are more impactful. As other groups within and beyond Dell consider adding or changing capabilities in their reverse supply chains, they may wish to follow a similar process to what we used to here to test assumptions by identifying the important factors and the best and worst case scenarios associated with these factors. We believe the approach outlined in this chapter would work well in a variety of reverse supply chain contexts.

Chapter 7. Reusable Packaging Case Study

The ARM model, as a high-level strategic planning tool, provides broad guidance on which reverse supply chain capabilities to enable, like a warranty stock replenishment recovery channel, at even how much inventory and of what quality to send to the channel. What it does not do, however, is provide guidance for optimizing the channel itself. When a supply chain designer inputs the revenue stream as well as the associated fixed and variable costs into the model, the model simply takes these numbers as given rather than providing any cost-reduction guidance. Of course, this is as intended to keep the model straightforward and its objectives simple; however, a gap remains. In this chapter, we introduce a supporting tool to the ARM model to help fill this gap: a reusable packaging cost/benefit analysis calculator. We describe the capabilities of the tool in the context of a case study we undertake with Dell's GSP and Finance groups.

7.1 Overview

For at least seven years, with avid support from its CEO, Dell has concerned itself with environmental responsibility, from offering customers free eco-friendly disposal of old computer parts before its competitors, locally sourcing innovative bamboo packaging, powering its operations with renewable energy, designing products that consume less power, and manufacturing them with partially from recycled materials [53]. Even within the reverse supply chain, Dell likes to be green. The example we focus on in our case study is the reusable packaging used by GSP when it performs warranty exchanges.

Dell initiates a warranty exchange when a customer experiences a problem with a part or product recovered under Dell's warranty policy. After support staff diagnoses the problem, typically over the phone by asking the customer questions or having him or her try various corrective measures, GSP ships the customer a replacement in a reusable box. The customer, upon receipt of the new item, places the faulty item in the same box and mails it back to Dell with the provided return shipping label. This closes the packaging loop. Assuming the box is in good condition, Dell will reuse it for a subsequent exchange.

The introduction of SFF devices provides GSP with a good opportunity to reevaluate their choice of packaging. The smaller form factor presents new packaging options—packaging tailored to the smaller form factor for smartphones or their main constituent parts, such as batteries. As we engaged GSP to consider the options, we quickly discovered inherent difficulties in comparing them. The main complicating factors are lifespan and shrinkage. We use *lifespan* to refer to the number of round trips a reusable package can make before it loses its integrity and must be recycled. We use *shrinkage* to refer to the proportion of customers who will not return the box after they receive their replacement items, either because they are delinquent, return the faulty item back a separate box, or are committing fraud. These factors complicate the cost/benefit analysis, as there are no straightforward formulas to take them into account.

7.1.1. Objective

In this case study, our objective is to accurately quantify the costs of various environmentally-friendly packaging options, and recommend the most cost-effective option to Dell. To mitigate the uncertainty associated with any assumptions in our analysis, we want to back the recommendations with a sensitivity analysis.

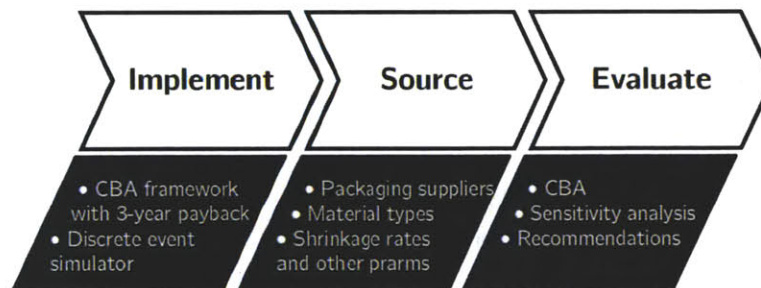
7.1.2. Approach

We define a three-step approach to meet our objective:

1. Implement the CBA tool. Work with the Finance group to determine the preferred approach to CBA analysis, and incorporate reusable packaging concepts such as lifespan and shrinkage into that approach.
2. Source packaging options. Investigate available packaging options from multiple suppliers, and collect information on their costs and lifespans.
3. Evaluate. Pull together the information from the first two steps to provide a thorough CBA, including a sensitivity analysis. Based on the data, issue a recommendation to Dell.

We summarize this approach in Figure 56.

Figure 56: Methodology of the reusable packaging case study



7.2. Implementation

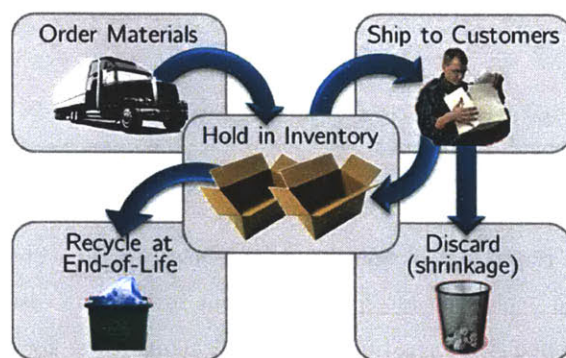
Dell Finance provides an Excel spreadsheet to assist groups throughout the company with their CBA calculations. They accommodate a baseline scenario—the current state—along with up to three alternative scenarios. By inputting investment costs and variable costs over time, the spreadsheet automatically calculates the NPV of the four options respective costs over a three-year horizon. The scenario with the lowest NPV is the least costly choice.

First, we attempted to define a formula to express the cost of a reusable packaging option over time. This proved problematic. Not only are there the costs of boxes and related materials, such as tape or ties, but there's the issue of when to order these supplies and how often to order them. The order fre-

quency depends on factors such as packaging lifespan, shrinkage rate, and turnaround time—how long it takes a package to make one loop from GSP to the customer and back. Ultimately, we determined a formula capturing all of this information would be hard to define or maintain, so we settled on a different approach.

As an alternative to a formula, we devise a discrete event simulator to model packaging materials flowing through the reverse supply chain. Figure 57 illustrates the design of this network. Material enters the system based on an order for packaging materials, and then loops through the system continuously between inventory and customers. At either point, the material may exit the system. The customer may never return the packaging (shrinkage); or, GSP may dispose of the packaging as it reaches its end of life (retirement). The simulator tracks the flow of materials at week-level granularity, meaning it processes 156 (52×3) discrete events, or periods.

Figure 57: Reusable packaging material network



We implement our simulator as a VBA macro within the CBA spreadsheet designed by Dell finance. We add a new worksheet entitled “Packaging Input,” shown in Figure 58. When a user fills out this worksheet, it the packaging costs into the existing CBA framework. This framework rolls up all costs and computes three-year NPV values per Dell’s specifications. Thus, anyone performing a CBA analysis at Dell may easily incorporate a reusable packaging component.

Figure 58: Screenshot of the reusable packaging worksheet within Dell's CBA spreadsheet. Input fields are denoted with a yellow background.

Reusable Packaging

3-Step Package Calculation Worksheet

INSTRUCTIONS: Follow these 3 steps instead of filling in the Fixed and Variable Cost sheets manually.

In step 3 below, quarterly ordering and material costs are calculated for you automatically using an reusable packaging simulator.

STEP 1: Describe the packaging alternatives

	Baseline	Scenario 1: Solid plastic	Scenario 2: Corrugated plastic	Scenario 3: Air bag	
One-time costs:					
Certification/testing	\$ -	\$ 5,000.00	\$ 5,000.00	\$ 5,000.00	Cost of "drop tests" that certify packaging is appropriate for the equipment. It will carry (if necessary).
Tooling, etc.	\$ -	\$ -	\$ 540.00	\$ -	Costs of initial tooling, as well as any other one-time setup costs.
Box:					
Material cost	\$ 0.49	\$ 2.99	\$ 4.36	\$ 0.30	Cost of an individual box.
Order quantity	3,000	2,500	3,000	2,500	The number of boxes that are shipped from the box manufacturer in a single order; must be 0 or greater.
Shipment cost per order	\$ 120.00	\$ 835.00	\$ 160.00	\$ 375.00	The cost a carrier charges to ship the order of boxes from the box manufacturer to Dell.
Lifespan (# uses)	5	60	10	5	Maximum number of times a box can be used; must be between 5 and 90; a value of 2 means the package is not reusable.
Materials:					
Material cost	\$ 0.06	\$ 0.03	\$ 0.06	\$ 0.10	Cost of materials that are used with a box (security ties, tape, outer envelope, etc.).
Order quantity	1	6,000	1	1	The number of items that are shipped from the materials supplier in a single order; must be 0 or greater.
Shipment cost per order	\$ -	\$ 20.00	\$ -	\$ -	The cost a carrier charges to ship the order of materials from the materials supplier to Dell.
Material quantity per box	1	2	1	1	The quantity of materials consumed per exchange; must be 0 or greater.
Miscellaneous:					
Shrinkage (% of customers not returning)	0.2%	0.2%	0.2%	0.2%	Percentage of customers that will not return their old equipment or will render the box unusable; must be between 0% and 100%; a value of 100% means no.
Inventory buffer (% of exchanges)	0.0%	0.0%	0.0%	0.0%	Additional box and material inventory desired, relative to what's needed to handle the average exchange volume; must be greater than or equal to 0%.
Turnaround time (weeks)	3	3	3	3	Number of weeks it takes a box to travel from Dell to the customer and back again; must be a whole number.

STEP 2: Specify the expected quarterly volume

	Year 1-Q1	Year 1-Q2	Year 1-Q3	Year 1-Q4	Year 2-Q1	Year 2-Q2	Year 2-Q3	Year 2-Q4	Year 3-Q1	Year 3-Q2	Year 3-Q3	Year 3-Q4
Exchange volume	7,000	8,400	11,000	13,000	15,000	18,000	22,000	27,000	32,000	38,000	45,000	55,000

STEP 3: Click the button to the right to run a simulation that will automatically determine optimal box and material order quantities for each quarter

	Year 1-Q1	Year 1-Q2	Year 1-Q3	Year 1-Q4	Year 2-Q1	Year 2-Q2	Year 2-Q3	Year 2-Q4	Year 3-Q1	Year 3-Q2	Year 3-Q3	Year 3-Q4
Baseline:												
Box	2	1	1	1	2	3	2	4	4	4	5	7
Materials	7,001	8,400	11,000	13,000	15,000	18,000	22,000	27,000	32,000	38,000	45,000	55,000
Total cost	\$ 3,600.06	\$ 2,094.00	\$ 2,350.00	\$ 2,370.00	\$ 4,080.00	\$ 5,910.00	\$ 4,500.00	\$ 7,980.00	\$ 8,280.00	\$ 8,640.00	\$ 10,650.00	\$ 14,430.00
Scenario 1 - Solid plastic:												
Box	2	-	1	-	-	1	1	1	1	1	1	2
Materials	5	5	3	5	5	6	7	9	11	13	15	18
Total cost	\$ 17,220.00	\$ 600.00	\$ 8,910.00	\$ 1,000.00	\$ 1,000.00	\$ 9,510.00	\$ 9,710.00	\$ 10,110.00	\$ 10,510.00	\$ 10,910.00	\$ 11,310.00	\$ 20,220.00
Scenario 2 - Corrugated plastic:												
Box	4	-	2	1	3	3	3	3	3	8	5	9
Materials	7,001	8,400	11,000	13,000	15,000	18,000	22,000	27,000	32,000	38,000	45,000	55,000

STEP 3: Click here to run simulator

A user performing a reusable packaging analysis follows the three steps listed on the worksheet:

1. Describe the packaging alternatives: input investment costs, box costs, materials costs, and miscellaneous information such as shrinkage rates for up to four scenarios. We describe each of these inputs in detail in Table 51.
2. Specify the expected quarterly volume of exchanges for the next three years.
3. Click a button to run the simulator. As output, the simulator indicates the quantity of box orders and materials orders to place in each of the next 12 quarters. Through Excel formulas, this information is translated into costs that are integrated with Finance's existing CBA framework, which roll up all the information to NPVs for each of the four scenarios.

For more details on the implementation of the reusable packaging simulator, consult Appendix C. We provide a source code listing in that appendix.

Table 51: Explanation of input fields for the simulation

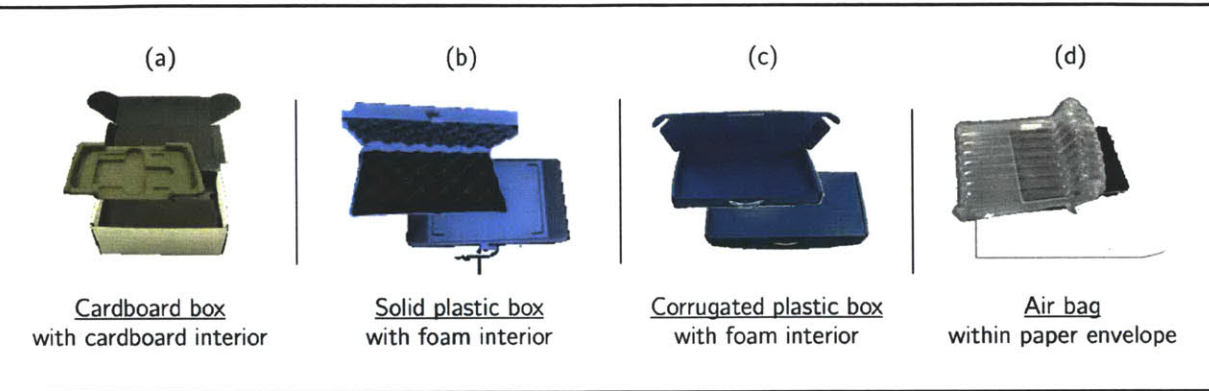
Category	Input field name	Explanation
One-time costs	Certification testing	The costs involved in certifying the packaging sufficiently safeguards its contents in transit. Typically, this includes “drop testing” to simulate rough manual or mechanical handling. Dell requires certification by an approved third party before it uses a type of packaging to transport product.
	Tooling, etc.	The costs involved in preparing manufacturing equipment to produce the packaging. Some packaging suppliers pass along these costs to Dell, especially if Dell requests customizes a package’s shape or color.
Box	Material cost	The unit cost of an individual box or container.
	Order quantity	The number of units the supplier ships in a single order.
	Shipment cost (per order)	The cost of shipping a single order to a GSP location.
	Average lifespan	The number of round trips the box will likely make before losing its structural integrity or physical attractiveness; at such a point, the box is discarded.
Materials	Material cost	The cost of single-use materials associated with the box, such as tape, ties, or envelopes.
	Order quantity	The number of units of material in a single order.
	Shipment cost per order	The cost to ship the materials from the supplier to a GSP location.
	Material quantity per box	The number of ties, envelopes, or other appropriate unit of measure required for a single warranty exchange round-trip. For example, if the material to close a box is a plastic tie, then two such ties may be needed: one to seal the box on its way to the customer, and another to seal the box on its way back to Dell.
Performance	Shrinkage	The percentage of customers who will not return the box to Dell. A rate of 1% implies that one in every 100 customers does not return the box.
	Inventory buffer	The percentage of box and material inventory to hold beyond the needs of current demand. 0% indicates no additional inventory and 100% indicates double inventory.
	Turnaround time	The number of weeks it takes a package, on average, to complete a round trip from GSP to a customer and back again
	Exchange volume	The number of exchanges expected to occur in each quarter for the next three years (12 values total).

7.3 Sourcing of Packaging Options

Our research and conversations with multiple packaging suppliers led us to the four reusable packaging options, which we show in Figure 59. The cardboard box with cardboard interior (option *a*) is Dell’s current choice for SFF warranty exchange packaging, and our baseline scenario. On average, it handles about three uses before wearing out. The solid plastic box with foam interior (option *b*) is preferred by some 3PLs who exchange SFF devices. Highly rugged, it handles 60 uses before wearing out, though the security ties it uses to latch the lid are not reusable. The corrugated plastic box with foam

interior (option *c*) is custom-sized to our specifications. In terms of reusability, it lies between the cardboard box and solid plastic box with an average of ten uses. Finally, the air bag inside a paper envelope (option *d*) is the cheapest of the bunch in terms of material cost; it matches the cardboard box at the low end of lifespan with 3 uses, however.

Figure 59: Reusable packaging options sourced from four different vendors: (a) cardboard box with cardboard interior, currently in use by Dell, (b) solid plastic box with foam interior, (c) corrugated plastic box with foam interior, and (d) air bag placed within paper envelope



7.4. Evaluation

With our baseline scenario and three alternatives identified, we describe the complete set of inputs we use to compare them in Table 52. As a reminder, we do not use actual data to preserve confidentiality. Notice that unlike the cardboard box, the three other packaging options must incur a \$5,000 initial expense for certification testing. This will be a significant hurdle for these options' viability; Dell's CBA policies use a 3-year horizon, however, which does provide a fair amount of time to pay back that investment. Also notice that, in the performance category, we level the playing field by using consistent choices for shrinkage, inventory buffer, and turnaround (TAT) time across all four scenarios. We base our shrinkage percentage and TAT time on GSP's experiences with other form factors.

Table 52: Simulation input parameters for baseline and the three alternative scenarios

		Baseline: Cardboard box	Scenario 1: Solid plastic box	Scenario 2: Corrugated plastic box	Scenario 3: Air bag
One-time costs	Certification testing	\$0.00	\$5,000.00	\$5,000.00	\$5,000.00
	Tooling, etc.	\$0.00	\$0.00	\$540.00	\$0.00
Box	Material cost	\$0.49	\$2.99	\$4.36	\$0.30
	Order quantity	3,000	2,500	1,000	2,500
	Shipment cost (per order)	\$120.00	\$835.00	\$160.00	\$375.00
	Average lifespan (# of uses)	3	60	10	3
Materials	Material cost	\$0.06	\$0.03	\$0.06	\$0.10
	Order quantity	1	6,000	1	1
	Shipment cost per order	\$0.00	\$20.00	\$0.00	\$0.00
	Material quantity per box	1	2	1	1
Performance	Shrinkage	0.2%	0.2%	0.2%	0.2%
	Inventory buffer	0.0%	0.0%	0.0%	0.0%
	Turnaround time	3	3	3	3
	Exchange volume	7,000 exchanges/quarter in the first quarter, with volume increasing by roughly 20% each quarter, ending at 55,000 exchanges/quarter			

7.4.1. CBA Results

With inputs specified, the reusable packaging simulation calculates, over three years, Dell will need to place 104 cardboard box orders, 16 solid plastic orders with 293 materials orders, 109 corrugated plastic orders, or 124 air bag orders. We show the NPVs of these scenarios in the column labeled *Expected* of Table 53. The current cardboard box packaging option turns out to be the cheapest, while the solid plastic and air bag options do not lag far behind. The custom corrugated plastic option is the outlier, at more than double the price.

To see whether the cardboard box option remains the best under different scenarios, we test a few of them: we try doubling the number of exchanges, doubling the turnaround time, and dramatically raising the shrinkage rate to 2%. We find that under these different scenarios, the results of which we also summarize in Table 53, the baseline scenario still fares best in all but one of them: doubled demand. Given its much longer lifespan, solid plastic appears to provide benefits in higher demand scenarios. We investigate this further with a sensitivity analysis.

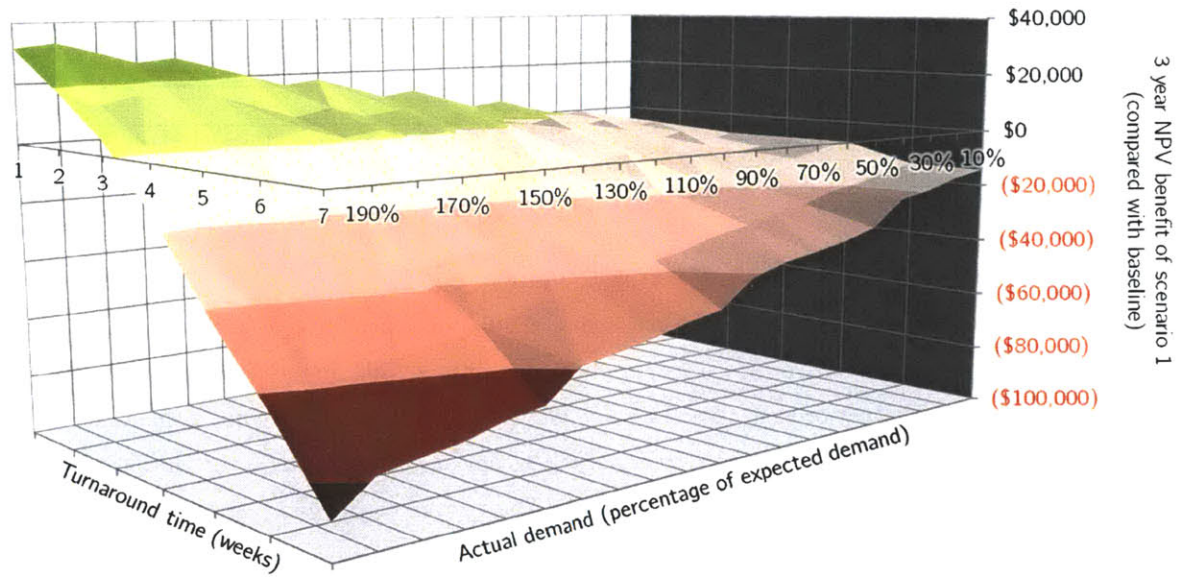
Table 53: Three-year NPVs of the four reusable packaging scenarios under different conditions

	Expected	Doubled demand	Doubled TAT	2% shrinkage
(a) Baseline: Cardboard box	\$61K	\$122K	\$63K	\$63K
(b) Scenario 1: Solid plastic box	\$64K	\$111K	\$100K	\$78K
(c) Scenario 2: Corrugated plastic box	\$160K	\$306K	\$188K	\$176K
(d) Scenario 3: Air bag	\$69K	\$127K	\$70K	\$70K

7.4.2. Sensitivity Analysis

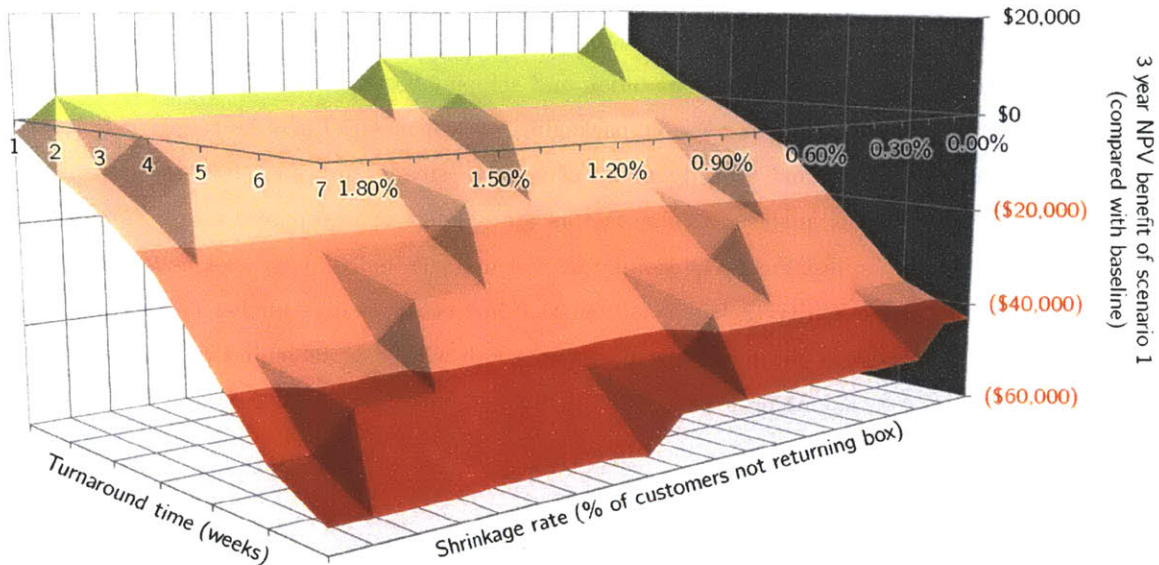
By running the simulator and performing the CBA calculations repeatedly under different scenarios, we explore the effect of varying demand on the preferred packaging option. Figure 60 shows an x-axis with demand ranging from 10% of the expected exchange rate in each quarter over the next three years to 200%, or double, the rate. On the z axis we vary TAT from 1 week to 7. We then plot the difference in NPV between the solid plastic scenario and the baseline scenario. Values that are positive (shown in green) indicate the solid plastic packaging is the preferred option. The majority of the graph is below the \$0 threshold, demonstrating that a high-volume, low-TAT environment is well suited to solid plastic packaging, but otherwise its longer lifespan cannot justify its higher per-box costs.

Figure 60: Sensitivity analysis of baseline vs. scenario 1 with actual demand and turnaround time factors



We look at this from another perspective in Figure 61. Here we examine shrinkage rate instead of volume but keep the other axes stable. Again, we find a limited range where the solid plastic packaging is appealing: low shrinkage rates with low TAT.

Figure 61: Sensitivity analysis of baseline vs. scenario 1 with shrinkage rate and turnaround time factors



7.4.3. Recommendations

We recommend Dell continue using its existing cardboard packaging. We cannot foresee a situation where the corrugated plastic or the air bag provide it with a better value, and the situations where the solid plastic box performs better is limited to situations where there is high volume, low TAT, and very low shrinkage. Each of these three dimensions has to be beyond our expectations for the solid plastic box to pay off.

With some additional analysis, we find that the solid plastic box does become less expensive when exchange volumes do not increase very quickly year to year, since, instead of requiring additional (expensive) plastic packaging to support new demand, Dell can leverage the longer lifespan and reuse its existing inventory at little cost. This demand profile is not likely, though. Dell will more likely continue growing its SFF volume or discontinue the form factor altogether; either scenario renders the plastic box unappealing.

7.5. Generalizable Insights

The solid plastic box supplier makes a compelling case for a five-month ROI on its product. That supplier's straightforward analysis leaves out several factors, however, including ordering costs, lifespan, and, critically, shrinkage rate. The popular web-based CBA tools we reviewed, such as the Reusables Cost Comparison Tool supported by the Reusable Packaging Association [54], also omit these factors. With the reusable packaging simulator and CBA spreadsheet, we incorporate these factors into the analy-

sis and place them in the context of Dell's volume estimates, yielding a surprising result that showed no ROI—and, in fact, tens of thousands of dollars in added costs—under most circumstances.

The generalizable insight here is that estimating reusable packaging costs is not as simple as it may seem. A straight comparison of box, material, and shipping costs, which would be sufficient for single-use packaging, is not sufficient for reusable packaging. When material may be reused, additional factors like TAT, shrinkage rate, and even dynamics of volume (steady, growing, shrinking, seasonal, etc.) play an important role in the analysis. These factors are difficult to incorporate into a simple formula, but with the benefit of our discrete event simulator, which tracks individual packages over a simulated three-year period, we can quantify their effects. As Dell and other vendors further embrace eco-friendly reusable packaging, we believe our simulation-based approach will be a viable and reliable way to assess the true ROI of various packaging options.

Chapter 8. Teardown Dashboard Case Study

In our fourth and final case study, we create a dashboard to monitor continuously the effectiveness of a central process in the reverse supply chain: the teardown process. With insights from this teardown dashboard, GRL and ARB can not only make necessary tactical changes to maintain high levels of net recovery, but also feed the information—such as updated teardown capacity limits and recovery estimates—back into the ARM model to periodically reevaluate the reverse supply chain design. In essence, then, the teardown dashboard provides a good tactical complement to the high-level teardown recommendations of the ARM model.

We begin the chapter with an overview of the teardown process at Dell. We discuss our objective and approach with respect to this case study, and then describe the collaborative process we took with members of GRL, proceeding from 1) mocking up the dashboard design, 2) aggregating data using Microsoft Access, and 3) implementing the user interface using Microsoft Excel. We conclude with generalizable insights from our experience.

8.1 Overview

A decision regarding whether to teardown a given system is more complicated than it may seem. There are only two possible choices, of course: either you teardown the system and recover value from its parts, or you do not, in which case you recovery value from the system as a whole, either by repairing it or selling it as-is. Knowing which choice is best requires knowledge of the future, though. Up front, you cannot be certain what value can be extracted from the parts—as some may be non-functional or missing, nor can you be certain what the cost of repairs might be—a system returned with a reason code of a faulty part may turn out to have no-fault found (NFF). Further, a priori the ultimate price a customer or third party will be willing to pay for either the parts, or for the integrated system is influenced by supply and demand, increasing obsolescence, and other factors. To complicate matters further, there may be internal demand among groups at Dell for a subset of the parts. Based on these factors, and more, how do you make the best decision?

Lin's 2009 work with the GRL team yields a decision-support tool to estimate the tradeoffs [29]. Based on models of historical net recovery for systems, forecasts of likely repair costs, and historical failure rates for parts, among other variables, the tool provides reverse supply chain operators with specific teardown recommendations regarding the quantity of systems to teardown in each product family over the next week (or month). He estimates that for every 1% improvement the tool can make towards determining the optimal number of systems to tear down, net recovery improves by about 0.25%. At Dell's return volume, the 1% improvement equates to over a million dollars in additional earnings.

With the assistance of a third-party software company, Dell is now integrating Lin's economic model into an automated teardown recommendation engine. This engine, while making more optimal decisions than the original manual approach, is not foolproof. With many considerations underpinning the

teardown decisions, some of which are based on forecasts of varying accuracy, GRL cannot afford to assume the model always makes the right tradeoffs. The earnings implications are too great. Therefore, in this case study we work with them to develop a teardown dashboard to monitor the tools recommendations over time. This allows GRL to change the tools input or algorithm as necessary, and also turn to the ARM model when the changes are great enough to warrant a reevaluation of the reverse supply chain design.

8.1.1. Objective

In this case study, we aim to identify the key metrics to monitor as part of the teardown process, collect the data necessary to calculate these metrics, and then present the data in an intuitive manner that effectively identifies anomalies worth investigating. To ensure the dashboard's rapid development and effective utilization over time, we impose some requirements on the implementation:

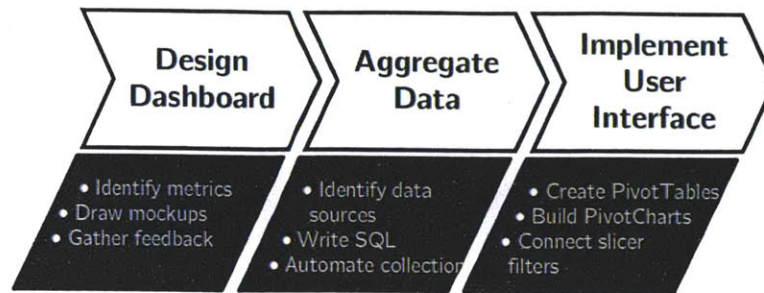
1. Use "commodity parts." Given limited IT resources, we implement the tool with software available to all stakeholders: applications in the Microsoft Office product suite.
2. Automate data collection and aggregation. We do not wish to create a burden on any GRL member by requiring manual data collection. Such a burden would likely inhibit the timely sharing of information, or may result in the dashboard being abandoned altogether.
3. Ensure the dashboard is sharable. The GRL team often distributes monthly scorecards via email or the SharePoint CRM system. We want the dashboard to harmonize with this paradigm. That means the dashboard interface cannot be bogged down with raw data.

8.1.2. Approach

We use a three-step approach, summarized in Figure 62, to conduct this case study:

1. Design the dashboard. In conjunction with GRL members, we identify the relevant metrics and roughly sketch the dashboard concept using fake data. We incorporate the feedback and repeat the process several times, honing in on a good design.
2. Aggregate data. We identify the specific data we need, then work with database domain experts to build the database commands to fetch it as needed. We build an infrastructure to fully automate data collection.
3. Implement the user interface. We bring the mockups to life using the actual data collected, leveraging some new features in Excel to produce interactive, filterable dashboard screens.

Figure 62: Methodology of the teardown dashboard case study



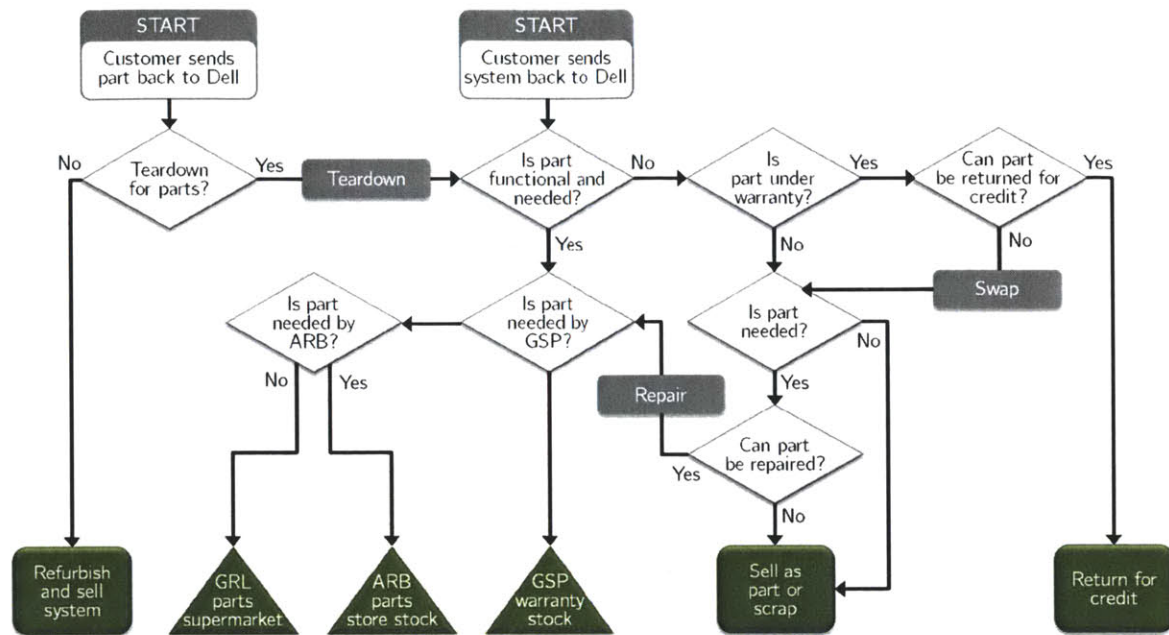
8.2. Dashboard Design

We begin the dashboard design process by mapping out a process flow for teardown. The results, slightly simplified, are shown in Figure 63. The process flow helps us not only hone in on some of the key metrics, but also reveal some of the stakeholders, other than GRL, with an interest in the teardown process. Based on the process flow diagram, we see two other stakeholder groups: ARB and GSP. ARB uses parts from teardown to help stock its online parts store. GSP uses such parts to help stock its inventory of parts for warranty exchanges. There are three more recovery channels: GRL's part supermarket for refurbishing other systems, and external avenues where parts are returned to suppliers for credit or sold through auctions or as scrap. Ideally, our dashboard would monitor opportunities and demand from all of these avenues to ensure teardowns are occurring at the appropriate frequency and, subsequent to teardown, that parts are flowing to the most lucrative recovery channels.

The process flow also helps us identify a possible structure for the dashboard. We see a logical breakdown of data into sections based on the flow of product:

1. Incoming systems. Returned systems are what feed the teardown process. We need to understand volumes and percent torn down by product family.
2. Parts. We need a breakdown of parts, but since there are so many parts we also need a way to group similar parts together. We can use Dell's *commodity* groupings, such as central processing units (CPUs), motherboards, keyboards, and so on, to collapse thousands of parts into just a few dozen commodities.
3. Inventory levels. We show the three parts inventories in the flow diagram as green triangles. We can show those inventory levels, by commodity, or, more granularly, by part, and show how the levels change over time.

Figure 63: Context of teardown in the flow of parts through the reverse supply chain

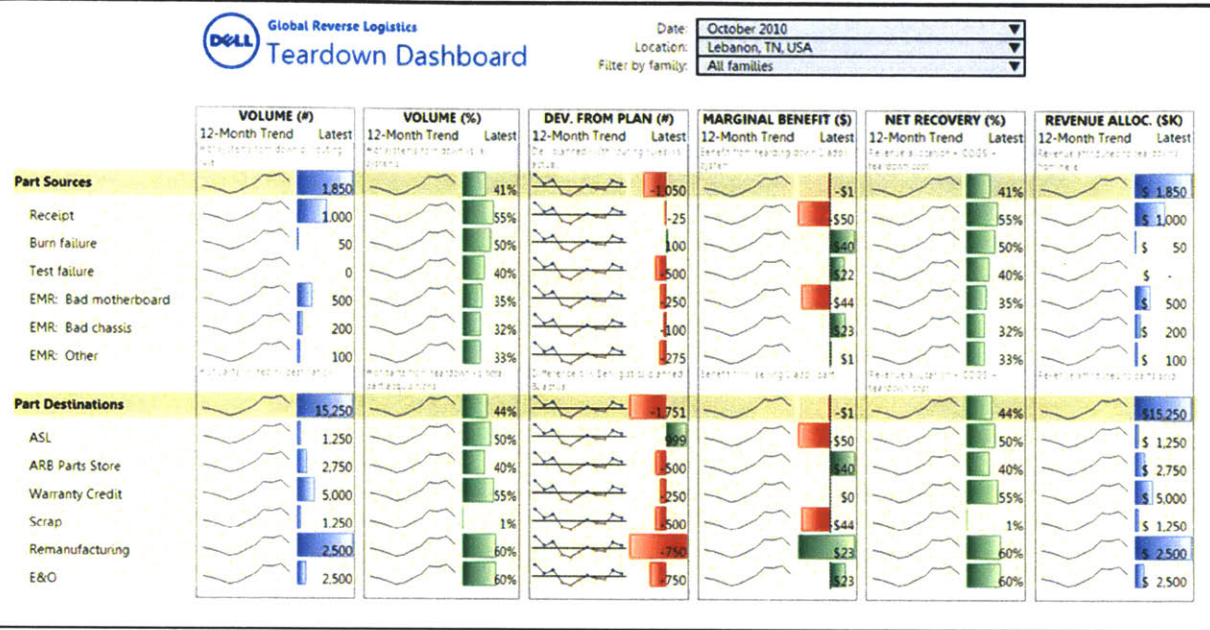


We pull these concepts together in our initial mockup, shown in Figure 64. We provide a filter at the very top to choose the date, location, and system families to analyze. The first half of the table shows statistics by complete systems, i.e. before teardowns, and the second half shows statistics by parts. On the system side, we subdivide our reporting by routing location, that is, where within the reverse supply chain the decision is made to teardown the system. The further along a system moves within the process, the more money is spent on it pre-teardown, but, at the same time, the more is known about. Dell's teardown decisions can be simple, such as *teardown a random 10% of systems in family A*, in which case the systems are routed to teardown from the arrival or "receipt" location. Alternatively, Dell may wish to select the desired 10% from the population experiencing a failure at a burn or EMR location, which might be indicative of a subpopulation that would be more expensive to get back into working condition. Still another option is to choose the systems from those with specific needs, like new chasses or motherboards, which are two of the more expensive components to replace due to not only material cost but the high labor cost involved. In truth, Dell may wish to pursue a variety of these routing decisions, so the dashboard endeavors to show the relative effectiveness of each one.

For systems and parts, our mockup provides detailed current state numbers as well as 12-month trends. We consider the trend data important to put the current state in context. The metrics we track include volume in quantity and dollar terms, the deviation between what was planned by the software and what actually happened on the plant floor, and estimates of the marginal benefit and net recovery of

the current activities. We knew some of the data would be difficult to capture, but at this stage we did not want to handicap our thinking by practical considerations.

Figure 64: Initial mockup for the teardown dashboard, tracking trends and current state for systems and parts by volume, relative volume, deviation from plan, marginal benefit, net recovery, and revenue allocation

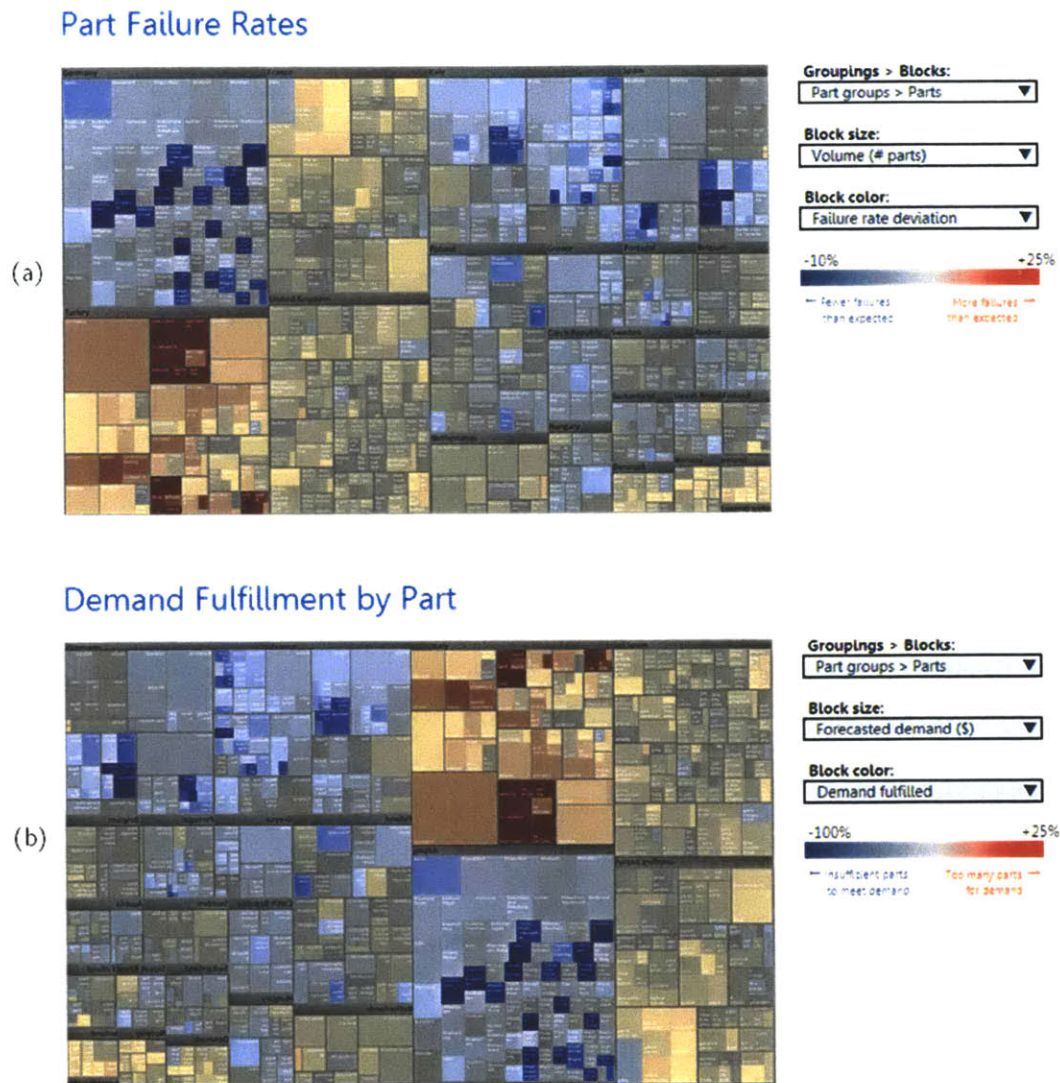


To supplement the primary dashboard screen, we propose squarified treemap visualizations, shown in Figure 65. A squarified treemap—first described by Wattenberg [55] and enhanced by Bruls et al. [56]—conveys three dimensions of information at once: hierarchical information based on the containment of successive rectangles, spatial information based on the relative sizes of these rectangles, and intensity information based on the color of each rectangle within a designated spectrum. In our part failure rate treemap shown in Figure 65(a), we group part rectangles into larger commodity rectangles, we use the volume of parts obtained from teardown to represent relative sizes, and we use color to indicate the number of part failures as compared to our forecasts: red indicating more failures than anticipated, and blue indicating fewer. This is an extensive amount of information to present simultaneously, for sure, however the squarified treemap form is quite easily readable. A person’s attention naturally gravitates toward the largest areas intensely lit in red or blue, and these are precisely the areas that deserve their focus. These are the areas where forecast error is simultaneously greatest *and* most impactful to the overall system. Presenting results in a table or chart, where a user may sort the information by one dimension or the other but not both simultaneously, is not as effective.

In Figure 65(b), we use a squarified treemap to show how well parts obtained from teardown are meeting demand. Rectangle containment, again, reflects the hierarchy of parts and commodities. Rectangle proportions now convey the relative total dollar-value of parts torn down (though another se-

lectable view is parts by quantity), and color indicates how well demand was met, with red indicating too few parts and blue indicating too many.

Figure 65: Squarified heatmap used to indicate major areas of concern with respect to (a) part failure rates, and (b) demand fulfillment by part. In both cases, rectangles size indicates overall volume and color indicates deviation from optimal: red for excessive failures in (a) and a shortage of parts in (b).



In the process of gathering feedback from stakeholders, we observe that mockups can provide a good basis for discussion. For some, a mockup is a good way to see the possibilities for a system, and to identify what might be missing. One of our key stakeholders noticed we missed a way to track inventory levels over time, for example, and asked we incorporate this into our implementation. We also saw drawbacks to the mockups. Some stakeholders had trouble separating the concept from the data; the dummy data led to confusion and at time sidetracked our discussion. On the whole, though, we thought the

mockups were worthwhile to share; they set expectations and also built excitement regarding the ultimate deliverable.

8.3 Data Aggregation

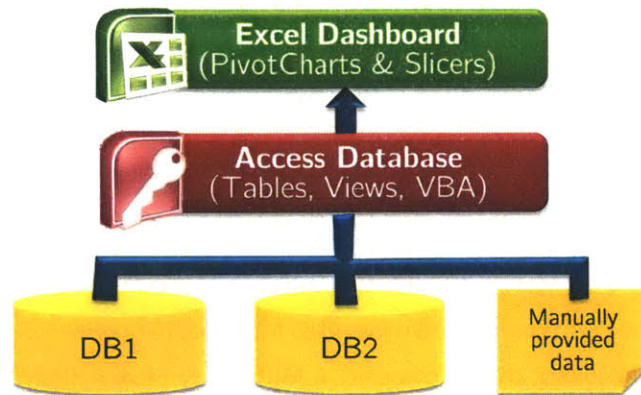
We work with Dell's database and reporting experts to identify the locations and techniques needed to extract the data. During this phase we ran into some challenges:

1. Some data are not available. We learned that some of the data is not currently tracked by Dell, and cannot be tracked without modifying existing supply chain processes. For example, to determine the ultimate disposition of a specific part, we would need to incorporate additional barcode scanning on the plant floor.
2. Some data are transitory. To keep database sizes manageable, Dell purges some tracking data from its reporting systems after a reasonable period of time elapses. For us, this means not being able to construct a full history of part flows; instead, we need to store our own summary of this data and allow it to accumulate in our dashboard over time.
3. Some data are held only by third-parties. With Dell contracting some supply chain functions to GENCO, and GENCO subcontracting further responsibilities such as teardown to other parties, some data, like part failure rates, is not available in Dell's systems. To get this information, we need to work with these third parties and establish a data feed link, through a scheduled file transfer protocol (FTP) drop, or some other mechanism.
4. Most data export activities are expensive. Given the volume of data and our goals of aggregating it into distinct time periods like weeks and months, we discovered the total extraction time would take at least a couple hours per week. This had ramifications on the dashboard architecture we ultimately devised.

We attempted to implement the dashboard exclusively in Excel, however we ran into severe performance problems as the size of the data tables grew. Further, with the long-running data exports, we realized end users would not be able to click a "Refresh" button and get updated data immediately. Instead, we employed a two-tiered architecture with a "back-end" database component and a "front-end" user interface component. We illustrate this design in Figure 66. The back-end is a Microsoft Access Database with tables, views, and VBA script to export data from a variety of data sources and maintain it in a structured way. The front-end is a Microsoft Excel spreadsheet, where PivotTables, PivotCharts, and slicer controls—powerful new filtering capabilities added in the 2010 edition of Excel—come together to show end users the desired metrics. The platform is designed such that the front-end can be decoupled from the back end. Thus, after a designated GRL employee refreshes the database on a weekly basis, he or she can sync the spreadsheet and then post it or email it, and the users of the spreadsheet can see the

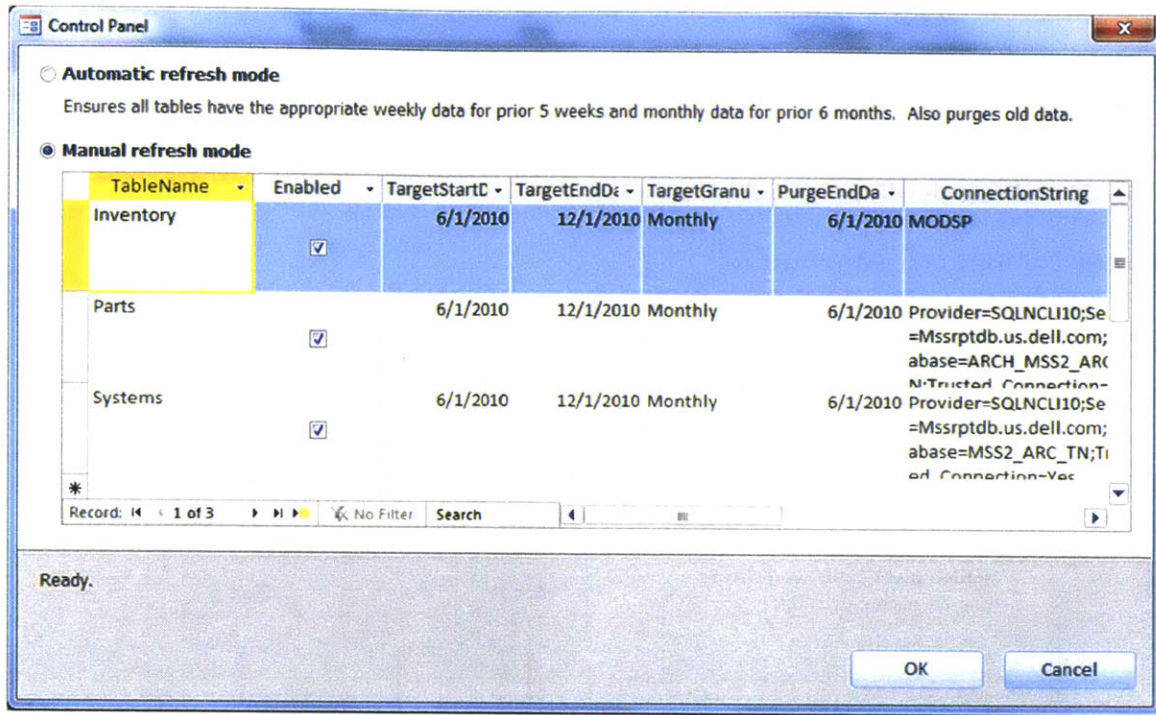
latest data as well as historical data, without connecting to the underlying data stores or even the associated Access database.

Figure 66: Dashboard architecture



Our data sources include a GRL database for system, a parts tracking in the reverse supply chain, and an inventory database used by GSP, and some manually provided, relatively static data as placeholders in lieu of arranging data drops with the various third parties. To manage these data sources, and provide an extensible way in which GRL can update these links, we provide a Control Panel interface to the Access Database. We show this interface in Figure 66, and provide relevant code listings in Appendix D. The code aggregates and extracts data from the data sources using structured query language (SQL) commands specified in the control panel and stores it locally in the Access database for integration with the Excel front-end. It also maintains the database by purging old data that is no longer needed.

Figure 67: Dashboard data control panel in manual refresh mode



8.4. User Interface Implementation

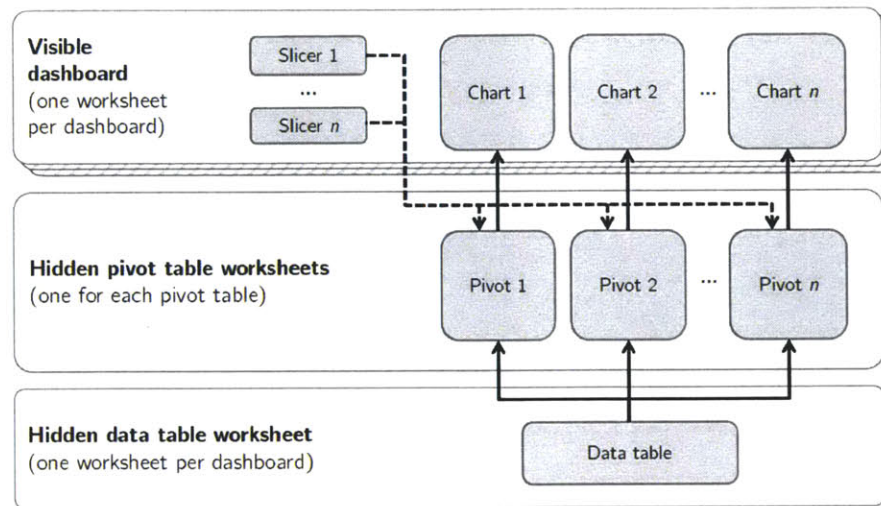
Combining our mockups with our database efforts, we are ready to pull it all together into a functional dashboard user interface. In our final round of mockups, the GRL team settled on three distinct dashboard worksheets: inventory levels, parts, and systems. The inventory levels dashboard provides a rudimentary inventory walk, showing inventory levels in each of the three main stockrooms over time—those stockrooms being GRL’s parts supermarket for remanufacturing, ASP’s parts warehouse for its online parts store, and GSP’s parts warehouse for warranty exchange stock. The parts and systems dashboard show parts obtained from teardown and systems torn down, respectively.

We assemble each dashboard using the features of Excel; no programming is required. We show an overview of the design in Figure 68. At the foundation of each dashboard is an Excel data table we link to a Microsoft Access view in the back-end layer. This simply means the data in the table can be periodically refreshed by pulling it from the Access database. As you recall, the Access database itself pulls data from a number of Dell data sources. We consider this two-step process vital in ensuring the performance of the overall system, as Excel is not well-suited to directly fetch and maintain data from a number of disparate data sources.

We then create a number of PivotTables, each backed by the same Excel data table. The PivotTables, which, like the data tables, are on hidden worksheets, power PivotCharts we place on a single

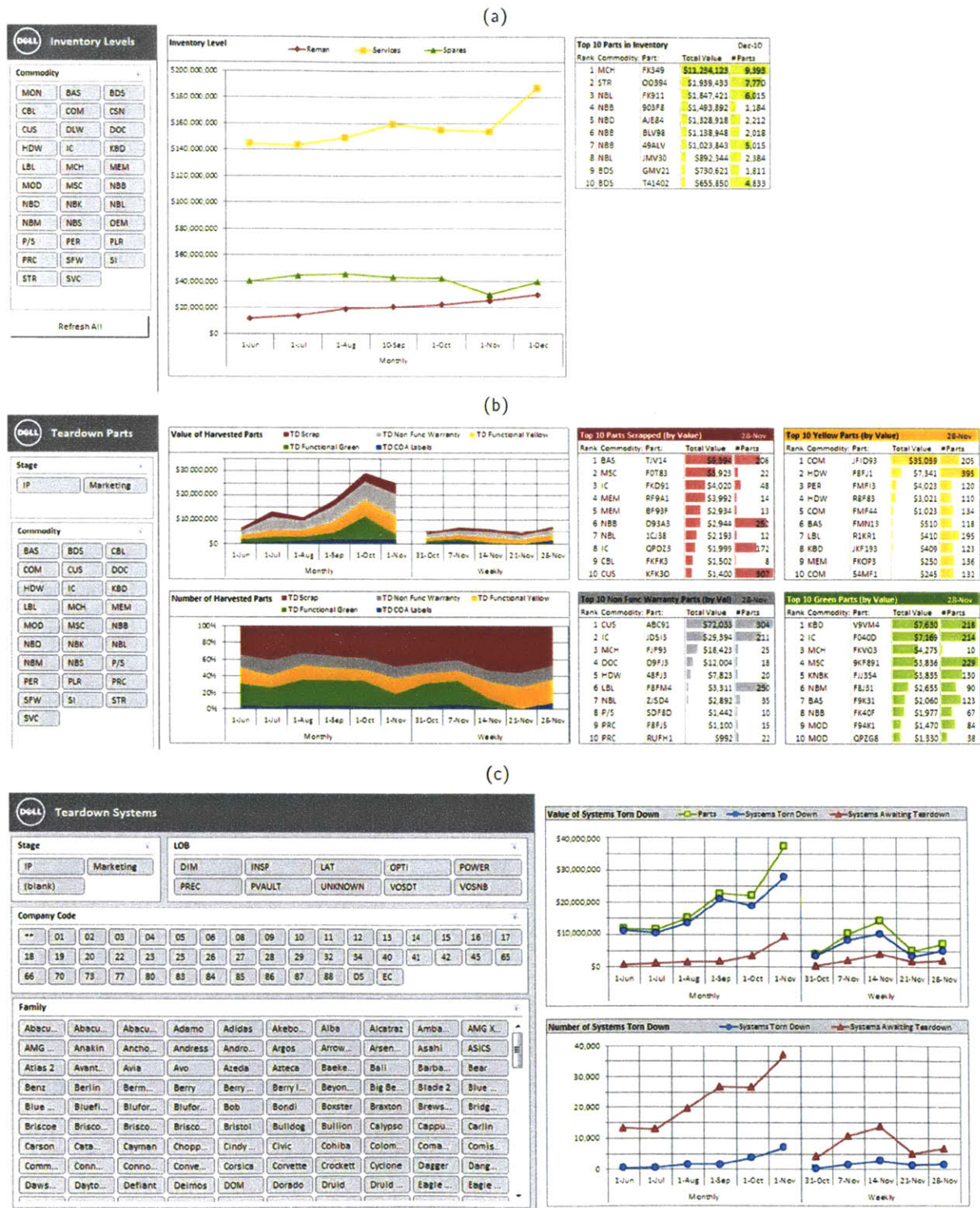
visible dashboard worksheet. (Excel requires each unique PivotChart to be backed by its own PivotTable.) Finally, we create filter controls called *slicers* with which users interact to select subsets of data along any available dimension. Using a new feature of Excel, we link each slider to *all* of the PivotTables driving the PivotCharts on the dashboard. With this arrangement, a single click on a filter updates all of the charts on the dashboard simultaneously. It provides an intuitive, responsive user experience that accommodates slicing and dicing of data in many ways.

Figure 68: Dashboard front-end architecture



We show the three completed teardown dashboards in Figure 69: (a) inventory levels, (b) parts, and (c) systems. The slicers, acting as filters, appear on the left. They are automatically populated with available filter choices based on data in the underlying PivotTables. To their right are the PivotCharts showing the key data. For most charts, at the request of GRL, we show data at two intervals of granularity: at the monthly levels for the last six months, and at the weekly level for the past five weeks. To the right of the charts we include several “top 10” and “bottom 10” lists reflecting, for the most recent period, parts with the highest cumulative value in inventory, scrapped, “yellow” (needing repair), credited under supplier warranty, and “green” (functional).

Figure 69: Current implementation of the teardown dashboard, showing (a) inventory levels across ARB, GSP, and GRL, (b) part inventory and trends, including top parts scrapped and reclaimed by value and count, and (c) systems torn down by number and value



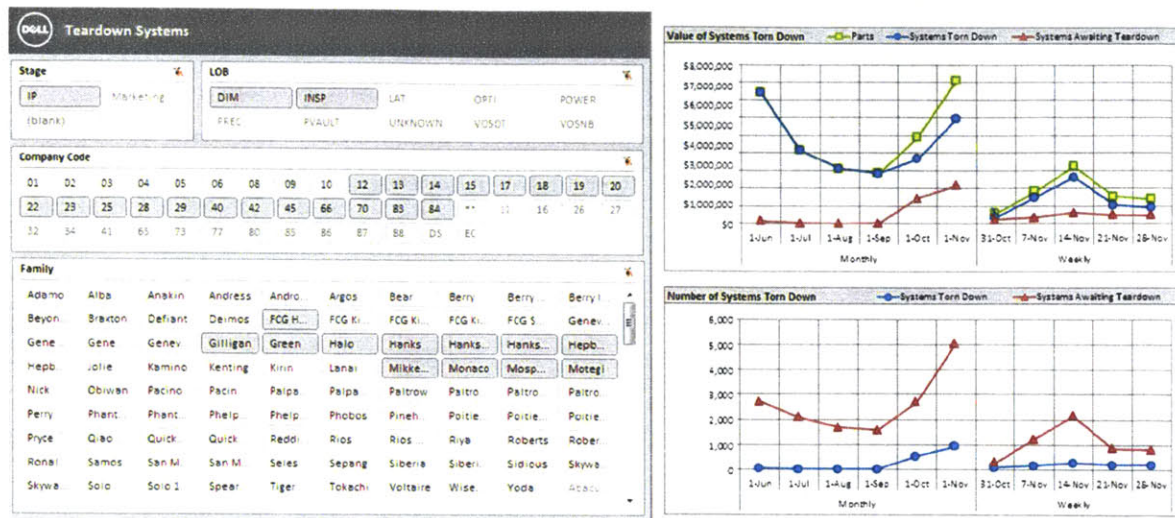
The data we show in these dashboards, which is admittedly straightforward, nonetheless provides value to the GRL team. Even before we finished implementing the version of the dashboard, the charts gave GRL some great insights into the current health of the teardown process. As we reviewed some of the charts with the team, we noticed two alarming trends: a recent spike in the number of parts being scrapped, and a growing queue of systems awaiting teardown. Before the dashboard, the team did not have easy access to this information, and therefore team members were not in a position to act. Now, armed with the dashboard, they took this information directly to the plant floor to investigate along with the 3PLs involved.

In addition to the power of the data visualization itself, a compelling feature of the dashboard is its filtering abilities. We showcase these abilities in Figure 70. Compare to Figure 69(c), which shows all data. Within the “Stage” filter of Figure 70, we have clicked the filter value *IP* (in production, for systems still available for sale as new through Dell’s website), and also some other filters. As we do so, not only do the charts update to reflect our selections, the filters affect each other. Multiple selections are supported. Each filter value can be in one of four states:

- A dark gray background indicates a filter value is currently selected. Data for it is shown.
- A light gray background indicates a filter is selected, but no data corresponding to it shown, as other selected filters exclude it.
- A white background with gray text indicates the same thing, but the filter is not selected.
- A white background with black text indicates a filter is not selected, but by selecting it, more data will appear on the charts.

GRL members consider these filters valuable. Instead of assembling multiple reports for IP and non-IP systems, for example, they can use a single dashboard and quickly create those reports on the fly, along with many others.

Figure 70: Demonstration of slicer-based filtering in action on a panel of a dashboard. As a filter is selected, the linked charts and the other filters react. Dark gray indicates a selection, white a deselection; light gray indicates no data available



Out of necessity, the actual dashboard is less comprehensive than the mockups. Some data simply is not collected or not easily accessible. Instead of postponing the rollout until every last data source is in place, however, we see value in providing a dashboard even with a subset of functionality. The dashboard fills a visibility gap, and as GRL members begin ramping up their teardown activity they are looking for a clear way of ensuring they manage the process optimally. Over time, we see the dashboard evolving as additional data sources become available and team members express interest in new charts or metrics.

8.5. Generalizable Insights

Our dashboard system shows the possibilities for automating recurring reporting needs without much IT support. We think the value of such a dashboard cannot be overrated, especially when it comes to monitoring a critical yet complex business process such as teardown. We found the GRL group in need of additional visibility without a management burden, and our generic dashboard framework provides this for the group. We believe other groups, both internal to Dell and external, can find use in it as well.

A key insight from this case study involves the process by which a dashboard is designed and deployed. We found mockups a good starting point. We feel it important to consider the ideal metrics up front, without regard for feasibility. It helps people identify what data are most important to run the business process, and can also set the bar for the longer-term picture of the dashboard. So long as user expectations are set appropriately, there is minimal harm. Watch for stakeholders who question the validity of data in the mockup stage; keep the conversation focused on metrics and design. Perhaps most importantly, bring in all the stakeholders early on. Since the dashboard will likely depend on data from

multiple groups, getting their joint ownership early on is critical. We found a process flow diagram helpful in identifying the stakeholders and initial metrics for the mockups, but a value stream mapping or database diagram may work just as well depending on context.

Two other points for a successful dashboard implementation: keep stakeholders in the loop throughout development, and do not hold back an implementation waiting for more data sources to become available. The dashboard is designed to support a process of continuous improvement. Set the expectation that the dashboard itself is an ever-evolving tool to help its users meet its objectives.

Chapter 9. Conclusions and Next Steps

Through four case studies, we have explored several systems to manage the reverse supply chain. The principal system, the Asset Recovery Maximizer (ARM) model, is a MILP-based strategic planning spreadsheet capable of recommending network designs, visualizing network flows, performing sensitivity analysis, and estimating financial metrics, while maximizing profitability. With ARM, we incorporate useful features from preexisting models while offering novel extensions, including multi-period flow analysis with price and cost depreciation considerations, a dynamically generated flow network based on user input, a straightforward interface requiring no special software, and several optimizations to keep the average-case performance to polynomial time, meaning results are often available within a few seconds.

We employ the ARM model in our first case study, in Chapter 5, to assess the financial impact of SFF returns in the U.S., determine the most influential factors affecting net income and recovery, and recommend a reverse supply chain design given this uncertainty. We find that in the baseline scenario, a fully-enabled reverse supply chain, complete with repair, teardown, and in-house resale capabilities improves Dell's current design by about 50%. When subject to a sensitivity analysis, we attribute volatility of this estimate primarily to overall return rate, with more units sold leading to higher profitability. We find that cost erosion is only a minor factor. We also find that the net income and net recovery metrics do not always move in the same direction, and that a key factor to net recovery turns out to be weighted average standard cost, with higher costs leading to lower net recoveries. Overall, this case study provides a good initial treatment of the SFF reverse supply chain, and the multi-million dollar benefit of robust reverse supply chain capabilities.

We then moved to a case study regarding Alienware repair in EMEA in Chapter 6. Here we started by validating the model against the (former) current state, where repairs were not taking place. Again, using the ARM model, we performed the same type of impact analysis and sensitivity analysis. The results indicated positive income across virtually all scenarios, with refurbished price and refurbished sales capacity identified as the two key factors in influencing net income and net recovery. Surprisingly, both price erosion and repair costs had little impact in the analysis. Based on our assumptions and forecasts, we recommended moving forward with enabling the repair capability. The EMEA GRL and ARB adopted these recommendations. Going forward, they also plan on applying the ARM model to repair scenarios for other products, and, in fact, to periodically reevaluate the efficiency of other aspects of their reverse supply chain operations.

In Chapter 7, we acknowledge the need for supporting tools to complement the ARM model in the context of a reusable packaging case study. The ARM model's input fields look for simple values for set-up costs and variable costs, but sometimes, as is the case with reusable packaging, determining these costs is not always straightforward. Accounting for the complexities of limited lifespan, shrinkage rate, and ordering costs, we devised a discrete-event simulator integrated with Dell's standard CBA tool to compare simultaneously up to four reusable packaging scenarios. Our results show discrepancies with

packaging suppliers' claims of a rapid ROI, and our recommendation to GSP was to continue with its current cardboard packaging supplier.

Through our final case study in Chapter 8, we highlight the importance of teardown. Teardown capacity and revenue estimation is another straightforward input in the ARM model that nonetheless requires some sophisticated thought by reverse supply chain managers. With a number of inputs affecting these decisions—such as part inventory levels, parts under warranty, repair costs, and revenue from system refurbishment—erring one way or the other can leave a reverse supply chain with lower a net recovery and excess inventory. We devised a teardown dashboard to monitor the current state while placing it within the context of trends. Most importantly, we provided a means by which the data is updated automatically, without IT support, and easily digestible with filtering and sharing capabilities. The dashboard system drives a continuous monitoring and improvement process, which in turns leads supply chain managers back to the ARM model to reevaluate the effectiveness of teardown against other possible transformations.

Cumulatively, we find these case studies fruitful at demonstrating the value of the ARM model to multiple functions at Dell globally. Based on the feedback we have received from members of GRL, ARB, and GSP, we believe the ARM model has, and will continue to foster better collaboration and sharing of best practices across these groups and regions, and also lay the groundwork for a more rigorous and timely evaluation of reverse supply chain capabilities.

9.1 Specific Recommendations to Dell

During our six-month engagement with Dell, we collaboratively developed the ARM model with a global, cross-functional team of employees. We also worked with employees to conduct the case studies. During this time, we learned much about the company from structural, political, cultural, and capability perspectives, as discussed in Chapter 3. Based on this experience, we offer some recommendations pertaining to the reverse supply chain:

1. Consider treating ARB as a standalone profit center. For accounting purposes, Dell structures ARB as a virtual P&L, partially beholden to the BUs policies and directives, such as BOM-matching standards of repair. Janse et al. suggest a shift in mindset, where a firm's ARB group operates independently as its own profit center [3]. With a change, ARB might be better incented to pursue cost-saving measures, such as lowering the reverse supply chain's P/D ratio similar to that of DFS's.
2. Simplify metrics and policies. Complex calculations may impede optimal decision-making. The net recovery metric, for one, may be misleading since its standard cost basis is not based on the LCM principle; it may end up underreporting the value of recovery or lead to inefficient behavior. Additionally, it does not always trend directionally with profitability. We demonstrate these discrepancies in our case studies. As an alter-

native, and in line with our previous recommendation and ARM model objective function, we suggest profitability (net income) as the key metric for ARB.

3. Align incentives with 3PLs. Dell has arrangement with many 3PLs for various reverse supply chain functions. Sometimes the ABC activities of these 3PLs discourage optimal routing through the supply chain. For example, an ideal place to route systems to teardown is at the point of a burn failure; however, this may not be the case if the cost of the failed activity must be incurred by Dell. There may be opportunities to reevaluate contracts in a manner mutually beneficial to Dell and 3PLs.
4. Institute preponement processes. As suggested by Guide, for high-value, shortly-lived products—which would include SFF devices—a more costly, less centralized “preponement” design may yield higher profits [25]. Dell should work with its retail partners to determine which reverse supply chain activities, such as light-touch repair and refurbished sales, are feasible at their distribution centers and retail outlets so as to shorten the turnaround time of refurbishment and cut out excess logistics costs.
5. Continue sharing best practices among groups and regions. To some extent the ARM model has helped bring different groups and regions together sharing best practices. Each region has a unique supply chain design, with best practices the other regions may find valuable. We encourage the groups to continue pursuing collaborative projects.
6. Consider the reverse supply chain in product design. Subramoniam demonstrates the value of products designed with the reverse supply chain in mind; such products can make disassembly, repair, and environmentally friendly disposal easier [21]. ARB, GRL, and GSP should be part of the product planning process to ensure these considerations are considered for the benefit of Dell’s overall profitability.
7. Periodically reevaluate the reverse supply chain using the ARM model. The ARM model’s flexibility lends itself to use in a variety of contexts—with specific configurations as well as entire form factors, and with specific countries and well as multiple continents. Given the ease with which scenarios may be run, and sensitivity analysis performed, the model can help a supply chain manager quantify an initiative, identify critical factors, and assess risk based on the most uncertain or volatile factors. Given rapid improvements in technology, new types of product designs, and continuous changes in the economics of various recovery channels, ARB, GRL, and GSP should find periodic use of the model to validate current structures and evaluate new ones a worthwhile activity. It provides a great opportunity to identify and pursue any latent but fleeting advantages in the marketplace ahead of competitors.
8. Avoid outsourcing strategic decision-making. As Fine cautions, “Among capabilities, this competency of selecting all others *is not to be outsourced!*” (author’s emphasis) [48]. By

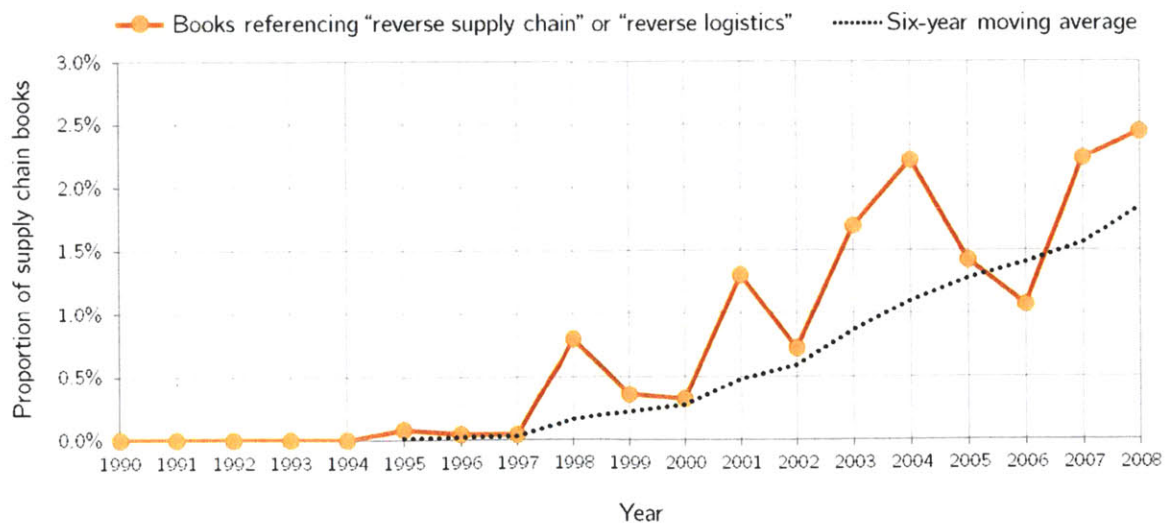
investing employees' time in developing and leveraging the ARM model, we believe Dell is demonstrating a commitment to keep such capabilities in-house, even as it looks increasingly towards 3PLs to execute its plans. We advise it to hold the line here, and, in other words, not turn operation of the ARM model over to its 3PL partners.

To put these recommendations in perspective, Dell's reverse supply chain demonstrates remarkable efficiency and capability, and is not only industry leading, but also world class. Our recommendations aim to further enhance its existing competitive advantages in this area.

9.2. General Implications

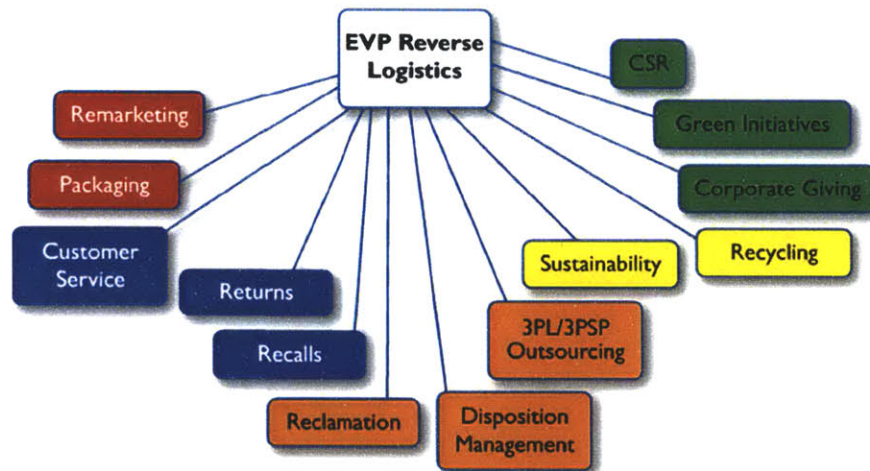
The U.S. alone faces hundreds of billions of dollars-worth of product returns annually [2], and sales from refurbished goods add up to more than \$50 billion annually [57]. Perhaps due to this impact, reverse supply chain as a field is growing in prominence, as shown in Figure 71. Nonetheless, the potential of fully realized reverse supply chains remained to be fully tapped. That is good news for firms like Dell that use the reverse supply chain for competitive advantage, not only in additional revenue streams from refurbishment, but in supplementing warranty stocks and part store inventory, complying with government regulation, and fostering customer goodwill through green initiatives. Giuntini and Gaudette consider the reverse supply chain the "next great opportunity in boosting U.S. productivity" [58]. They see 85% energy savings in remanufacturing, compared with new product manufacturing, along with benefits to the consumer: prices that are 30-40% less than new. The reverse supply chain can also open up new markets to firms [9].

Figure 71: Prominence of reverse supply chain concepts in supply chain literature in n-grams. Of the books scanned by Google between 1990 and 2008, proportion of those with the n-gram phrase "reverse supply chain" or "reverse logistics" of the ones with "supply chain" or "logistics" [59]



With reverse supply chain operations appearing as a new frontier in firms' quests for temporary competitive advantage, management systems such as the ones we propose in this paper are important to ensure operational efficiency and continuous improvement. Vick, President of the Reverse Logistics Association, notes that well-executed reverse logistics operations have the capability of adding 5% profit to a company's bottom line [60]. To do this requires systems such as the ARM model, but also a corporate structure to ensure the cross-functional, global teams collaborate on these systems' recommendations. Given its strategic value, Vick proposes a top-level role at firms that centralized reverse supply chain responsibilities of customer service, service logistics, after-market supply chain, sustainability initiatives, and CSR departments, as shown in Figure 72. We believe our management systems would work especially well with firms embracing a role such an *Executive Vice President of Reverse Logistics*, or, alternatively, firms such as Dell with a sufficiently collaborative environment to enable successful cross-functional working groups.

Figure 72: The functions of Vick's proposed "EVP of Reverse Logistics" role [60]. The different colors correspond to the fragmented reporting structure of these functions in most firms today.



9.3. Remaining Research Questions

We proposed several specific enhancements specific to the ARM model in Section 4.7.2. Here, are we conclude, we offer a summary of two of the more challenging outstanding questions warranting more research in the field of closed-loop supply chain design. First, consider cannibalization. One common question we receive is how the ARM model, or any model, accounts for cannibalization—that is, the loss of revenue from new product sales as customers turn to refurbished products instead, such as those offered on the Dell Outlet or eBay. For the ARM mode, the simple answer is “by incrementing the variable cost input field by the cannibalization amount.” The more complex answer of determining such an amount is not known. Cannibalization may not be relevant if the new and refurbished markets are disjoint; or, it

may promote customer loyalty and, eventually, fuel new product sales; or, it may severely hinder new product sales. In 2008, Guide et al. called for more research into this area [8], and we still see that need.

Second, consider preponement. We see promise in the preponement model proposed by Guide [25], but with a twist that makes it more cost effective: leveraging a firm's the network of distributors and retailers to minimize their own capital costs. Dell already does this for its EOL collections—consumers can return their Dell-branded machines to any Staples store, for example—but not yet for other reverse supply chain functions such as refurbishment or resale. More research into which reverse supply chain components to decentralize, along with perhaps support in the ARM model to analyze the tradeoffs, might boost net recovery significantly.

Appendix A. Glossary

We divide the glossary into three sections:

1. Dell-specific terminology. Like any large company, Dell has its own lexicon and acronyms for various groups and processes. In this section, we define the Dell-specific terms used in this text.
2. ARM model terminology. Here we define terms that have special meaning in our ARM model implementation.
3. Closed-loop supply chain terminology. Here we define the broad concepts that appear across CLSC research.

A 1 Dell-specific Terminology

AGOP (*approved global operations process*) – Process of the Operations and Technology corporate function by which project proposals affecting reverse supply chain capabilities are prioritized and scoped. Once approved, projects are shepherd through their lifecycle via the PRP process.

ARB (*asset recovery business*) – Group within in the consumer segment of Dell charged with minimizing returns and obtaining the highest net recovery for returned, excess, and obsolete equipment.

ASP (*America's service providers*) – Regional division of GSP focusing on North and South America, principally the United States.

BU (*business unit*) – Market-based segment of Dell's product development, sales, and marketing capabilities. The four BUs of Dell are large enterprise, public, services, and SMB and consumer.

core team – Group of cross-functional representatives, who collectively represent all the stakeholders involved in a particular project. Core teams are usually defined during an *AGOP* process, but are fluid and may grow or shrink as the project evolves during the *PRP* process.

corporate function – Top-level structure at Dell, alongside the BUs, providing capabilities to all the BUs. The six corporate functions are finance, HR, legal, marketing, operations and technology, and strategy.

DAO (*Dell Americas Operation*) – Regional segmentation of the globe covering North America and South America.

DFS (*Dell Financial Services*) – Independent subsidiary of Dell, within the Dell Finance corporate function, offering financing options to customers of all segments, as well as leasing options for a variety of Dell- and non-Dell-branded hardware.

digital nomad – One of the work classifications for Dell employees, an intermediate classification between *on-site* and *remote*. Digital nomads typically work from home, but have access to “hotel” cubes and storage lockers where they can check-in for an on-site workday. Dell created this classification in 2010, matching competitors such as HP who have cut costs with similar initiatives.

- ELT** (*executive leadership team*) – Twelve-member body consisting of the officers of Dell: the CEO along with the heads of Dell’s four business units, six corporate functions, and the enterprise product group. Dell’s overarching strategy and directives originate from the ELT.
- GRL** (*global reverse logistics*) – Group within the Operations and Technology corporate function of Dell that manages logistics related to the reverse supply chain globally, as well as asset recovery functions in APJ.
- GSP** (*global service providers*) – Group within the Services business unit providing customer assistance and warranty support globally.
- PRP** (*phase review process*) – Phase-gate process within the Operations and Technology corporate function by which all supply-chain enhancement projects are managed. The five phases are define, plan, develop, launch, and sustain. To progress from one phase to the next requires passing a *gate*: a formal review, during which consensus must be obtained that previously established exit criteria has been met. See also: *SDLC*.
- SORT** (*systems optimization routing tool*) – Excel spreadsheet by Lin to make tactical decisions on the quantity of systems to tear down to maximize net recovery [29]. Decisions are driven by comparing, at the margin, the net income of refurbishment versus that of using a systems’ constituent parts.
- three pillars** – The three long-term strategy goals of Dell: client reinvention, eDell, and best value solutions. Each project at Dell aims to drive progress with respect to at least one of these pillars.

A 2 ARM Model Terminology

- ARM** (*asset recovery maximizer*) – The optimization model we describe in Chapter 4, so-named because its MILP features a profit-maximizing objective function based on recovering as much value as possible from returned products (assets).
- depreciation profile** – Series of percentages that include the period-by-period price or cost decline of a product over time. The ARM model allows a user to specify two depreciation profiles: one for price, and another for cost. In electronics, price erosion is typically attributed to increasing obsolescence over time, whereas cost erosion is typically attributed to volume discounts and improved capabilities by suppliers. By default we assume price erosion of 1%/week [26], and cost erosion of 0.25%/week [50].
- downgrade** – Process by which inventory of a particular type is re-classified as inventory of an inferior type. The ARM model downgrades products automatically as an optimization to route product through the reverse supply chain optimally. For example, if no recovery channels are available for new products, it will automatically reclassify it as refurbished inventory. See also: Figure 31.

modeling scenario – Description of a situation for which a user seeks a recommendation. The scenario includes one or more focal questions, a choice of product granularity, and a choice of region or country. When using the ARM model, we recommend beginning by considering the modeling scenario, which provides a good frame of reference when providing inputs or interpreting recommendations.

recovery channel – Final node in a reverse supply chain where systems, or their constituent parts, are ultimately sold or disposed of. Channels include third parties, such as via an auction process; other groups within the firm, such as replenishing warranty stock; or internally within the reverse supply chain, such as replenishing parts for refurbishment. Compensation from a recovery channel may take the form of immediate revenue (an outright purchase), deferred revenue (a revenue-sharing arrangement), or a fee/cost, such as that for disposal. *Example: Dell Outlet.*

rework activity – Intermediate node in a reverse supply chain where some improvement is made on WIP. *Examples: repair, teardown, ODM/OEM exchange.*

S&P (*software and peripherals*) – Designation of non-system parts that may or not be part of a BOM, including software applications and computer system accessories, such as monitors, printers, keyboards, mice, joysticks, and external hard drives. Dell’s reverse supply chain includes repair capabilities for a subset of S&P: mainly monitors and printers.

sensitivity scenario – An automatic modification of a baseline *modeling scenario*, where one to five inputs are adjusted, performed by the ARM model during a sensitivity analysis. A typical sensitivity analysis will involve various permutations of several input variables, yielding hundreds of sensitivity scenarios.

standard cost – See *COGS*.

transformational activity – See *rework activity*.

A 3 Closed loop Supply Chain Terminology

1PL (*first-party logistics provider*) – Firm or individual with a need to transport material from one location to another. *Example: Dell.*

2PL (*second-party logistics provider*) – Firm owning some means of transportation to ship material from one location to another. *Example: FedEx.*

3PL (*third-party logistics provider*) – Firm providing outsourced logistics services to a 1PL by coordinating activities among 2PLs. *Example: GENCO.*

4PL (*fourth-party logistics provider*) – Firm involved in the selection and management of 3PLs on behalf of a 1PL. *Example: Deloitte.*

AAIA (*automobile aftermarket industry association*) – A 23,000+ member association of repair shops, stores, and distributors in the \$281 billion U.S. automobile aftermarket [61]. Daugherty exports its members’ use of IT in 2004 [14]. See also: *RLA*.

- ABC** (*activity-based costing*) – Approach where activities, such as teardown or L1 repair in the reverse supply chain, are set at a unit price incorporating both variable and fixed costs. Some IPLs arrange ABC-based work contracts with 3PLs.
- APJ** (*Asia Pacific & Japan*) – Region of the world defined by some firms for administrative and accounting purposes, which includes countries in the continents of Asia and Australia, including China, India, Singapore, South Korea, Australia, and Japan. See also: *EMEA*.
- AWP** (*awaiting parts*) – inventory designation for systems destined for repair currently awaiting replacement (s) for defective or missing parts, which may be on order
- B&B algorithm** (*branch and bound algorithm*) – Computer algorithm used to find optimal solutions to linear and non-linear programs, especially ones which are NP-hard. The algorithm subdivides the solution space a hierarchy into “branches.” As it searches the solution space, it keeps track of the best solution found so far—the local optimum solution—and it excludes, or “prunes,” branches known to have inferior solutions, thus quickly narrowing the search space and improving the amount of time necessary to find the best solution. The ARM model uses this algorithm via the Solver add-in to generate its recommendations.
- BBN** (*Bayesian belief network*) – Network graph representing probabilities used by decision support systems accounting for uncertainties, such as Shevtshenko’s system [30].
- BER** (*beyond economic repair*) – A product or part requiring repairs where the cost of repair cannot be recouped through any recovery channel. In this case, the optimal activity is something other than repair—perhaps selling as-is or dismantlement. The state is transient, based on current market conditions.
- BOM** (*bill of materials*) – Hierarchical listing of all the subcomponents and parts which constitute a complete system. At Dell, the BUs typically require GRL to refurbish systems to match existing BOMs, limiting their degrees of freedom in choosing parts.
- bounty** – Rebate a mobile phone carrier provides to a retailer for enrolling a customer in a long-term service contract with the carrier, which reflects the customer acquisition cost and value of the contract over its term, typically of 2 years. Bounties are what make free phones, or those with sharply reduced prices, possible. They are by far most popular in the U.S., as most people in EMEA and APJ do not opt for long-term contracts. Retailers negotiate their own terms with carriers, but typical bounties are \$400+ for new activations, \$250+ for family plan activation, and \$300+ for upgrades. The average bounty is \$350+.
- CBA** (*cost/benefit analysis*) – Financial approach by which various competing alternatives are compared on the basis of cash flows, which are typically discounted to net present value (NPV). Dell performs CBAs using a 3-year planning horizon. We perform a CBA with reusable packaging in Chapter 7.

CLSC (*closed-loop supply chain*) – Comprehensive view of the supply chain consisting of both forward and reverse components. Firms with CLSCs, such as Dell and Caterpillar, engage in rework, re-manufacturing, and refurbishing activities that allow product to flow between manufacturer and customer multiple times.

CMS (*content management system*) – A system of organizing data, such as documentation, among multiple people, along with processes by which the data may be created, editing, and deleted. CMS systems are typically software applications available through a network. *Example: Microsoft SharePoint.*

CODP (*customer order decoupling point*) – the inventory buffer in a supply chain at which a forecast-driven (push) flow of materials becomes a demand-driven (pull) flow. *Ex:*

- A supply chain beginning with a CODP makes/builds to order (MTO/BTO).
- A supply chain terminating near a CODP makes/builds to stock (MTS/BTS).
- A supply chain with a CODP in its middle assembles to order (ATO).

COGS (*cost of goods sold*) – The value of an item in inventory based on all the variable and appropriate fixed costs that went into its manufacture. At Dell, a simplified form of COGS, known as *standard cost*, is sometimes used; it indicates the price Dell paid for the product from an OEM/ODM.

CRA (*customer return authorization*) – Permission that a firm grants to a customer to return a product, typically after verifying the return adheres to the firm's return policy, which may limit the time in which the return is permissible, or the reasons for which the return may occur. At Dell, when direct customers receives CRAs, they also receive shipping labels so the products are directed to the appropriate reverse logistics 3PLs. Retailers, on the other hand, are integrated with Dell's IT system to automate the CRA process.

DFD (*design for disassembly*) – Methodology by which reverse supply chain considerations are incorporated into the design of a product. Special considerations may include using materials that are easier to recycle, or designing modularly with components that are shared across products.

EMEA (*Europe, Middle East, and Africa*) – Region of the world defined by some firms for administrative and accounting purposes, which includes those in Europe, the Middle East, and Africa. See also: *APJ*.

EMR (*electromechanical repair*) – Activity of repairing electronic devices, such as laptops and smartphones, by replacing or fixing defective parts.

EOL (*end of life*) – Duration of time during which a product is considered obsolete or no longer is capable of providing its intended value to its user.

ERP (*enterprise resource planning*) – processes and IT systems providing comprehensive and forecasting data across the supply chain

- FGP** (*fuzzy goal programming*) – A variant of linear programming where multiple objectives, such as cost effectiveness and environmental impact, are expressed as decision variables that are then weighed against each other as the objective function. Tsai and Hung use this approach to devise a GSC [39].
- genetic algorithm** – Heuristic computer algorithm used to find near-optimal solutions to complex, NP-hard problems—such as optimization formulations—in polynomial time. The algorithm is modeled on the process of genetic evolution. It works by defining a series of feasible solutions, then progressively selects the most “fit” of these solutions (that is, those closest to optimal) and creates additional solutions through “mating,” a solution intermixing process. Min uses this algorithm with various reverse supply chain formulations [37] [38].
- GSC** (*green supply chain*) – A supply chain designed with environmental externalities in mind, not simply tangible profit maximization. These considerations include the use of green suppliers, environmentally friendly or recycled materials, and safe disposal of byproducts or EOL materials.
- ILP** (*integer linear program*) – Linear program where all decision variables are constrained to be whole numbers.
- IMEI number** (*international mobile equipment identity number*) – A 15 or 17 digit number used by a mobile service provider, to uniquely identify a telephony device, such as a mobile phone or laptop when it connects to the network. *Example: 12345678-123456-1-12.*
- IMSI number** (*international mobile subscriber identity number*) – A number, up to 15 digits long, used for authentication and billing purposes when a telephony device, such as a mobile phone or laptop, connects to a mobile network. *Example: 310150987654321.*
- LFF** (*large form factor*) – Shorthand reference to a class of electronics devices—typically 3 feet or more in any particular dimension—that, due to their size, often require special assembly lines in manufacturing facilities and special EMR stations in repair facilities. See also: *SFF*.
- logistics** – Management of flow of goods from raw materials at their point of origin to finished goods the possession of end customers. The field is quite broad, encompassing 2PL, inventory management, and packaging activities. See also: *reverse logistics*.
- LP** (*linear program*) – Formulation that expresses some objective in terms of decision variables, to which optimal solutions are sought so as to maximize or minimize said objective in the presence of zero or more constraints that place boundary conditions on the variables. The objective function and constraints must be expressed linearly with respect to the decision variables, otherwise the formulation is a non-integer program (NLP). LPs take a number of forms depending on the nature of the constraints, such as *ILP* and *MILP*; NLPs do too, such as *INLP* and *MINLP*. LPs are generally solved using a specific algorithm such as simplex or B&B, sometimes with the help of a heuristic such as genetic or simulated annealing, especially when the formulations’ complexities are NP-hard. The ARM model we present in this text is based on an MILP formulation.

macro – See *VBA script*.

MILP (*mixed-integer linear program*) – Linear program where some decision variables are constrained to be whole numbers.

MINLP (*mixed-integer non-linear program*) – Optimization formulation where the constraints, objective function, or both are non-linear with respect to the decision variables, and a strict subset of the decision variables are constrained to be whole numbers.

MRP (*material requirements planning*) – IT system dedicated to managing the production schedule and inventory levels in a manufacturing process. In the reverse supply chain, uncertainties related to part quality and quantity make usage of such a system problematic [17].

net recovery – Metric by which the value recovered from a customer return is measured, typically excluding fixed costs. Also can be used to measure the performance of the reverse supply chain as a whole, typically including fixed costs. Net recovery is a percentage expressed as income received from the returned asset divided by the total cost of the asset, which includes its standard cost plus any transformation activity costs plus, optionally, fixed costs. At Dell, standard cost is based on the current costs of the constituent materials in the product; elsewhere, the LCM cost may be used. *Example: Selling a refurbished system with a standard cost of \$100 and \$50-worth of repairs for \$125 has a net recovery of $125 \div (100 + 50) = 83\%$.* Across industries, net recoveries are often well below 100%.

netbook – A lightweight, inexpensive personal computer resembling a laptop—with a fold open design featuring a keyboard, track pad, and screen—typically with a screen size of 8 to 12 inches diagonally, featuring always-on internet connectivity via a mobile interface similar to that used by mobile phone. Netbooks are typically subsidized by wireless carriers in a manner similar to mobile phones, also typically with 2-year service contracts for the data bandwidth provided by the carrier. At Dell, the reverse supply chain sometimes strips netbooks of their mobility capabilities during the repair process in order to resell the hardware as “minis,” short for mini-laptops.

network flow formulation – Also *flow network formulation*. A type of LP formulation that involves routing items from source nodes to destination nodes via arcs, such that all items from the source flow to the destination, in the presence of capacity constraints on the arcs, such that the overall objective of minimizing the cost of flow along the arcs are minimized. The ARM model formulation is a variant of the network flow formulation with several extensions, including one to add a time dimension to the network.

NP-hard (*non-deterministic polynomial-time hard*) – Class of problems/formulations including those which, in the worst case, cannot be solved faster by any known algorithm than one whose running-time grows exponentially with respect to a linear increase in the number of inputs, such as the number of network nodes. A classic NP-hard problem is the traveling salesman problem. Many of the models we present in Chapter 3 are also classified as NP-hard.

- NPV** (*net present value*) – Indication of the value, today, of cash inflows and outflows occurring in the past and/or future based on the idea that there is an opportunity cost of capital, also known as the “time value of money,” measured by a potentially variable discount rate. When comparing opportunities, converting cash flows to NPV provides a standardized basis for comparison: the flows are collapsed to single numbers that can be compared directly.
- NVA** (*non-value-added activity*) – In contrast to a value-added (VA) activity, an activity considered wasteful because it does not enhance a good or service in a manner for which a customer would be willing to pay. *Example: holding inventory.*
- ODM** (*original design manufacturer*) – OEM who also assumes partial or complete design responsibility for a product. *Examples: Foxconn with respect to the Dell Aero, or Qisda with respect to the Dell Streak.*
- OEM** (*original equipment manufacturer*) – Manufacturer who assembles a product, or components thereof, on behalf of another firm and marketed with that firm’s brand. *Example: Flextronics.*
- P/D ratio** – Ratio of production lead time to delivery/fulfillment lead time, the two parts of the supply chain (or reverse supply chain) occurring before and after the *CODP*, respectively. Higher ratios indicate greater exposure to forecast error and higher inventory holding costs.
- PBA** (probability bound analysis) – Technique for incorporating uncertainty into a model using cumulative distribution functions (CDFs), as by Hamza et al. in their reverse supply chain model of total cost of ownership (TCO) [16].
- preponement** – In contrast to the cost-saving *postponement* (or deferment) model of the forward supply chain, this technique, devised by Guide et al. [25], is based on classifying product returns in the reverse supply chain as quickly and close to the source as possible, so as to repair and return to the market the most viable product as quickly as possible. Since preponement models cost more than centralized alternatives, the technique is preferable only in situations with high price erosion and expensive products, where rapid turnaround time is critical.
- RDMF** (*remanufacturing decision-making framework*) – defined by Subramomiam et al. [21].
- RF testing** (*radiofrequency testing*) – A type of electromagnetic compatibility (EMC) test to ensure a radiofrequency-emitting device, such as a mobile phone, is operating correctly. RF testing is a required capability of reverse supply chains engaged in mobile phone or netbook repair.
- RFI** (*request for information*) – Standard form of business communication where a firm issues a non-binding request, generally publicly yet anonymously, seeking information from third parties concerning their capabilities in a particular area. Based on responses, the firm decides on an action to take next, which may include engaging with one or more of the respondents. The *RLA* provides a forum by which members may post RFIs for reverse logistics services.

RFQ (*request for quote*) – A standard form of business communication where a firm requests pricing information on one or more specifically delineated activities, typically publicly and non-anonymously. It evaluates responses based on their prices, service levels, risk factors, and other characteristics.

ROI (*return on investment*) – When expressed as a point in time, indicates the point at which an investment in a good, service, or process has been recouped via cost savings, additional revenue, or both. When expressed as a ratio, indicates the change in net income relative to and resulting from the initial investment; an ROI at or over 00% indicates the investment has paid for itself.

reverse logistics – Specialty of *logistics* focused on the flow of goods from the end customer backwards towards a point of origin. Includes the process of processing returns, transforming them in some way, such as through repair, remanufacturing, disassembly, or disposal, and, ultimately, recovery of value through various recovery channels such as resale or warranty exchanges.

reverse supply chain – Specialty of supply chain management focused on the processes involved in managing product returns. Commonly used synonymously with *reverse logistics*.

RFID (*radiofrequency identification*) – System of asset location tracking based on small tags/labels affixed to the assets that emit a unique code when subjected to a radiofrequency from a radiofrequency reader/interrogator. RFID systems can be used in the reverse supply chain to track inventory, providing accurate metrics such as WIP in real-time.

RLA (*reverse logistics association*) – Association of thousands of 3PLs, ODMs, OEMs, branded and retail companies providing and/or seeking reverse supply chain services [62].

SDLC (*software development lifecycle*) – A generic term for any phase-gated process by which firms develop software. Phases may include planning, designing, development, testing, and maintaining. *Examples: waterfall, agile.*

SFF (*small form factor*) – Shorthand reference to a class of electronics devices smaller than a laptop; at Dell, this is generally understood to include smartphone and tablet devices, which may have screen resolutions up to 10 inches. See also: *LFF*.

SIM card (*subscriber identity module card*) – Module on a removable card of mobile devices—including phones and laptops—based on GSM (Global System for Mobile Communications) technology, which uniquely identifies the user of the phone by an IMSI number.

simplex algorithm– Computer algorithm by which linear programs are solved. It finds a basic feasible solution to the problem, then pivots along permutations of the solution until the optimal one is found. The average case running time of the algorithm is polynomial, even in the two corner cases where 1) a solution may not be possible, or 2) there is no maximum or minimum because the solution is unbounded.

simulated annealing – Heuristic computer algorithm offering near-optimal solutions to complex, typically NP-hard problems in polynomial time, such as optimization formulations. The algorithm, inspired by the physical process of cooling, generates random solutions and attempts to find better ones through slight permutations from these starting points. As the algorithm progresses—or “cools”—the starting solutions become less random, honing in on a local or global optimum. Used by Pishvae et al. in a reverse supply chain model [40].

SMB (*small and medium businesses*) – Also SME (small and medium enterprises). Segmentation of business delineated by a set maximum number of employees, typically 500 or fewer.

Solver – Extension to Microsoft Excel, developed by Frontline Systems, with capabilities for solving various types of linear and non-linear optimization problems expressed in terms of spreadsheet cells and formulas. A version with limited capabilities is bundled with all version of Excel, with premium versions available for purchase from Frontline Systems at additional cost.

SQL (*structured query language*) – Standardized programming language by which data is loaded, transformed, and extracted from relational databases. We use SQL in our teardown dashboard implementation.

tablet – Lightweight, small form factor personal computer, typically consisting of a 10 inch screen or smaller with pen or touch interface. *Examples: Dell Streak, Dell Latitude XT2, Apple iPad.*

tabu search – Heuristic computer algorithm similar to *simulated annealing*. The principal difference rests with the locality of the search. While simulated annealing looks broadly first and hones in on candidate solutions, tabu search looks locally first, adding moves it makes to a “taboo” list. For efficiency, it prunes redundant search paths reflected in this list.

TCM (*total cost management*) – Also *total competitiveness management*. Philosophy and framework described by Association for the Advancement of Cost Engineering (AACE) International to estimate and control the costs of a project throughout its lifespan.

teardown – Activity by which a system, such as a laptop or phone, is dismantled into its constituent parts, with the parts being consumed for various purposes such as remanufacturing, part sales, or warranty credits. Teardown is a valuable transformation when the value of the parts individually exceed the value of the system as a whole.

TEM (*total environmental management*) – Management philosophy and practice of environmental consideration throughout the lifespan of a project or product.

TQM (*total quality management*) – Management philosophy and practice of quality checks and continuous improvement throughout the design and manufacturing processes of a product.

TRM (*total risk management*) – Management philosophy and practice of risk minimization throughout the lifespan of a project.

VBA script (*Visual Basic for Applications script*) – Programming language used to automate tasks in software applications including the Microsoft Office suite. Individual subroutines or functions written in this language are also known as *macros*.

WAP (*weighted average price*) – Average price of a product or component, weighted by volume produced, available, or forecasted.

WEEE (*Waste Electrical and Electronic Directive*) – European law, effective February 2003, mandating obligations among electronic device manufacturers to accept and safely dispose of their products from E.U. customers at the products' end of life.

WIP (*work in progress*) – Material within the supply chain that is part-way through a transformative process, typically residing in an assembly line or queue. In the reverse supply chain, WIP includes non-functional computers awaiting parts (see *AWP*), fully functional systems awaiting teardown, and refurbished systems yet to be kitted. Also known as *in-process inventory*.

Appendix B. ARM Model Source Code

This appendix supplements the ARM model we describe Chapter 4. A number of macros power the model, including simple ones that change the visibilities of worksheets or interact with the Frontline Systems' Solver add-in. In this appendix, we present source code on the other, non-trivial macros: network flow drawing, benefit-per-box calculations, and sensitivity analysis.

B.1. Network Flow Diagram Drawing

The code in Listing 1 creates, colors, and sizes arcs—called *connectors* in the Excel object model—connecting statically placed network nodes representing the returns, rework, and recovery components of a reverse supply chain. It also labels these static nodes appropriately based on names the user has supplied on the input screen.

Listing 1: Network flow line generator macros

```
1
2 Private Const iNodeRightSide As Integer = 4
3 Private Const iNodeLeftSide As Integer = 2
4 Private Const iMaxLineWeightInPixels As Integer = 20
5
6 ' Clears and redraws the connectors of the network flow diagram based on current outputs
7 Public Sub DrawFlow()
8     Application.ScreenUpdating = False
9
10    ' Delete old connectors
11    Dim i As Integer
12    With Me.Shapes
13        For i = .Count To 1 Step -1
14            With .Item(i)
15                If .Connector Then .Delete()
16            End With
17        Next
18    End With
19
20    Application.ScreenUpdating = True
21    DoEvents()
22    Application.ScreenUpdating = False
23
24    ' Determine the max volume flowing through any arc in the reverse supply chain
25    Dim flow As Range
26    flow = FormulationSheet.Range("Flow")
27    Dim maxValue As Double
28    maxValue = WorksheetFunction.Max(FormulationSheet.Range( _
29        flow.Cells(, flow.Columns.Count + 2), _
30        flow.Cells(flow.Rows.Count, flow.Columns.Count + 2)))
31
32    ' Label each of the network nodes appropriately
33    Dim nodes As Range : nodes = NodesSheet.Range("NodeNames")
34    Dim n As Range
35    For Each n In nodes.Cells
36        Dim nodeID As String : nodeID = n.Cells(, -1).Value
37        FlowSheet.Shapes("Rect" & nodeID).TextEffect.Text = n.Text
38    Next
39
40    If maxValue > 0 Then
```

```

41      ' Connect the nodes to show transformation flows
42      For i = 1 To flow.Rows.Count
43          With flow.cells(i, flow.Columns.Count + 2)
44              If .value > 0 Then
45                  DrawConnector(FormulationSheet.cells(.Row, 1),
46                              FormulationSheet.cells(.Row, 2), .Value / maxVal)
47              End If
48          End With
49      Next
50
51      ' Connect customers to nodes to show return flows
52      Dim returns As Range
53      returns = NodesSheet.Range("ReturnStreams")
54      For i = 1 To returns.Rows.Count
55          With returns(i, returns.Columns.Count + 1)
56              If .value > 0 Then
57                  DrawConnector("O", .cells(1, -8), .Value / maxVal)
58              End If
59          End With
60      Next
61  End If
62
63  Application.ScreenUpdating = True
64 End Sub
65
66 ' Draws a single connector from one node to another at the given size
67 Private Sub DrawConnector(ByVal startNodeID As String, ByVal endNodeID As String, _
68                           ByVal pctSize As Double)
69     Dim startShape As Shape : startShape = FlowSheet.Shapes("Rect" & startNodeID)
70     Dim endShape As Shape : endShape = FlowSheet.Shapes("Rect" & endNodeID)
71
72     With Me.Shapes.AddConnector(msoConnectorCurve, 0, 0, 0, 0)
73         .ConnectorFormat.BeginConnect(startShape, iNodeRightSide)
74         .ConnectorFormat.EndConnect(endShape, iNodeLeftSide)
75         .Line.Weight = iMaxLineWeightInPixels * pctSize
76         .Line.ForeColor = IIf(endNodeID >= 20, endShape.Fill.ForeColor, _
77                               startShape.Fill.ForeColor)
78         .Line.Transparency = 0.4
79         .Line.EndArrowheadStyle = msoArrowheadTriangle
80         .Line.EndArrowheadLength = msoArrowheadShort
81     End With
82 End Sub

```

B.2. Benefit-per-box Calculations

The ARM model attempts to produce a benefit-per-box calculation for each rework activity and recovery channel, whether or not that activity/channel is currently enabled or utilized. As shown in Listing 2, the macro to perform these calculations runs the optimization algorithm for each activity/channel after tweaking the input in some way to calculate the benefit of enabling, disabling, or reducing its cost, as appropriate, after which the code resets inputs to their original values.

Listing 2: Benefit-per-box calculations

```

1
2 Public Sub CalculateCostSensitivities()
3     Dim i As Integer
4     Dim inputs As Range
5     Dim outputs As Range
6     Dim origNetIncome As Double
7     Dim totalVolume As Double
8     Dim reworkVolumes()
9     Dim recoveryVolumes()
10
11     origNetIncome = Range("NetIncome").Value
12     totalVolume = WorksheetFunction.Sum(Range("RecoveryVolumes"))
13     reworkVolumes = SaveInputs(Range("ReworkVolumes"))
14     recoveryVolumes = SaveInputs(Range("RecoveryVolumes"))
15
16     ' Calculate benefit-per-box for each rework activity
17     inputs = InputSheet.Range("ReworkInput")
18     outputs = RecommendationSheet.Range("ReworkResults")
19     For i = 1 To inputs.Rows.Count
20         outputs(i, outputs.Columns.Count) = CalculateBenefitPerBox( _
21             inputs(i, inputs.Columns.Count), _
22             inputs(i, 4), _
23             inputs(i, 5), _
24             outputs(i, 1), _
25             outputs(i, 4), _
26             reworkVolumes(i), _
27             origNetIncome, _
28             totalVolume)
29     Next
30
31     ' Calculate benefit-per-box for each recovery channel
32     inputs = InputSheet.Range("RecoveryInput")
33     outputs = RecommendationSheet.Range("RecoveryResults")
34     For i = 1 To inputs.Rows.Count
35         outputs(i, outputs.Columns.Count) = CalculateBenefitPerBox( _
36             inputs(i, inputs.Columns.Count), _
37             inputs(i, 3), _
38             inputs(i, 4), _
39             outputs(i, 1), _
40             outputs(i, 4), _
41             recoveryVolumes(i), _
42             origNetIncome, _
43             totalVolume)
44     Next
45
46     ' Reset inputs and outputs
47     ArmModel.RunBaselineScenario("")
48     RecommendationSheet.Activate()
49     FormulationSheet.Visible = xlSheetHidden
50 End Sub
51
52 Public Function CalculateBenefitPerBox(ByVal capLimitCell As Range, _
53     ByVal costCell As Range, ByVal setupCostCell As Range, ByVal optionName As String, _
54     ByVal volumeCell As Range, ByVal optionVolume As Object, _
55     ByVal origNetIncome As Double, ByVal totalVolume As Double)
56
57     Dim oldInput As Object : oldInput = capLimitCell.value
58     Dim oldCost As Object : oldCost = costCell.value
59     Dim oldSetupCost As Object : oldSetupCost = setupCostCell.value
60     Dim isCapLimitSupported As Boolean : isCapLimitSupported = oldInput <> "Not supported"
61     Dim isZeroCost As Boolean : isZeroCost = (oldCost + oldSetupCost = 0)
62

```

```

63 If optionVolume > 0 Then
64     ' We're currently using this option. Try not using it to see the difference.
65     If isCapLimitSupported Then
66         capLimitCell.value = 0
67     Else
68         costCell.value = 999999
69         setupCostCell.value = 999999
70     End If
71     If ArmModel.RunBaselineScenario("Calculating what happens with " _
72         & optionName & " disabled...") Then
73         CalculateCostSensitivity = (origNetIncome - Range("NetIncome").value) _
74             / totalVolume
75     Else
76         CalculateCostSensitivity = "Required"
77     End If
78 ElseIf capLimitCell.value = "0" Then
79     ' We're currently not using this option. Try using it to see the difference.
80     capLimitCell.value = ""
81     If ArmModel.RunBaselineScenario("Calculating what happens with " _
82         & optionName & " enabled...") Then
83         CalculateCostSensitivity = (Range("NetIncome").value - origNetIncome) _
84             / totalVolume
85     Else
86         CalculateCostSensitivity = "Infeasible" ' Should never happen
87     End If
88 Else
89     ' This option is not constrained, and yet is not being used.
90     If isZeroCost Then
91         ' It cost $0 and we're not using it. Nothing will get us to use it.
92         CalculateCostSensitivity = 0
93     Else
94         ' Try reducing all costs to $0. See if we use it then.
95         costCell.value = 0
96         setupCostCell.value = 0
97         If ArmModel.RunBaselineScenario("Calculating what happens with" _
98             & optionName & " at zero cost...") Then
99             If volumeCell.value = 0 Then
100                 CalculateCostSensitivity = 0
101             Else
102                 CalculateCostSensitivity = (Range("NetIncome").value - origNetIncome) _
103                     / totalVolume
104             End If
105         Else
106             CalculateCostSensitivity = "Infeasible" ' Should never happen
107         End If
108     End If
109 End If
110
111 capLimitCell.value = oldInput
112 costCell.value = oldCost
113 setupCostCell.value = oldSetupCost
114 End Function

```

B.3. Sensitivity Analysis

The sensitivity analysis macros shown in Listing 3 generate each combination of input values, run the optimization model for these sensitivity scenarios, and store the corresponding results in a data table for analysis in a connected PivotTable and PivotChart.

Listing 3: Sensitivity analysis macros

```
1
2 Private sensitivityTable As ListObject
3 Private outputLocation As Range
4
5 Sub RunScenarios()
6     sensitivityTable = SensitivitySheet.ListObjects("Sensitivity")
7
8     ' Name columns based on inputs and outputs
9     Dim cell As Range
10    Dim i As Integer : i = 1
11    For Each cell In SensitivitySheet.Range("SensitivityInputs,SensitivityOutputs")
12        sensitivityTable.ListColumns.Item(i).Name = String(i, " ")
13        i = i + 1
14    Next
15    i = 1
16    For Each cell In SensitivitySheet.Range("SensitivityInputs,SensitivityOutputs")
17        If cell.value <> "" Then
18            sensitivityTable.ListColumns.Item(i).Name = cell.value
19        End If
20        i = i + 1
21    Next
22
23    ' Save original inputs
24    Dim originalInputValues()
25    Dim inputCells As Range : inputCells = InputSheet.Range("AllInputs")
26    originalInputValues = SaveInputs(inputCells)
27
28    ' Reset results table
29    Dim outputRowIdx As Integer
30    outputLocation = sensitivityTable.Range.cells(2, 1)
31    Call Range(sensitivityTable.Range.cells(2, 1), _
32        sensitivityTable.Range.cells(sensitivityTable.Range.Rows.Count, _
33            sensitivityTable.Range.Columns.Count)).Clear()
34    Call sensitivityTable.Resize(Range(sensitivityTable.Range.cells(1, 1), _
35        sensitivityTable.Range.cells(2, sensitivityTable.Range.Columns.Count)))
36
37    ' Run through scenarios
38    Call RunScenariosOnInputColumn(1)
39
40    ' Reset inputs
41    Call LoadInputs(inputCells, originalInputValues)
42
43    ' Show scenario results
44    FormulationSheet.Visible = xlSheetHidden
45    SensitivitySheet.Range("ScenarioMsg") = "Last calculated on " & _
46    SensitivitySheet.Select()
47    SensitivitySheet.Range("ScenarioMsg").Activate()
48    Call SensitivitySheet.PivotTables("Sensitivity").RefreshTable()
49 End Sub
50
51 ' Recursively invoked subroutine to build scenarios from every desired value from each
52 ' input column
53 Sub RunScenariosOnInputColumn(ByVal inputColIdx As Integer)
```

```

54 Dim inputName As String : inputName = sensitivityInputs.cells(1, inputColIdx)
55 Dim low As Double : low = sensitivityInputs.cells(2, inputColIdx)
56 Dim high As Double : high = sensitivityInputs.cells(3, inputColIdx)
57 Dim increment As Double : increment = sensitivityInputs.cells(4, inputColIdx)
58 If increment = 0 Then increment = 1
59 Dim loops As Integer : loops = (high - low) / increment + 1
60 Dim x As Integer
61 Dim i As Double : i = low
62
63 ' Loop through each increment of the desired value from low to high
64 For x = 1 To loops
65     DoEvents()
66     If i > low Then
67         Dim j As Integer
68         For j = 1 To inputColIdx - 1
69             outputLocation.value = outputLocation.cells(0, 1)
70             outputLocation = outputLocation.Next
71         Next
72     End If
73
74     ' Apply target value to all cells with the given name
75     If inputName <> "" Then
76         Dim cell As Range
77         For Each cell In ActiveWorkbook.Names(inputName).RefersToRange
78             cell.value = i
79         Next
80     End If
81     outputLocation.value = i
82     outputLocation = outputLocation.Next
83
84     ' Capture recommendations with this value, or loop recursively if there
85     ' are more inputs
86     If sensitivityInputs.Count = inputColIdx Then
87         ' Run optimizer and generate recommendations
88         Call ArmModel.RunBaselineScenario("Calculating results (row " &
89             & outputLocation.Row & ") ... Please wait...")
90         Dim sensitivityOutput As Range, output As Range
91         For Each sensitivityOutput In sensitivityOutputs
92             If sensitivityOutput.value <> "" Then
93                 If (sensitivityOutput.cells(2, 1).HasFormula) Then
94                     outputLocation.value = sensitivityOutput.Cells(2, 1)
95                 Else
96                     outputLocation.value = ActiveWorkbook.Names( _
97                         sensitivityOutput.value).RefersToRange.Value
98                 End If
99             Else
100                 outputLocation.value = ""
101             End If
102             outputLocation = outputLocation.Next
103         Next
104         outputLocation = outputLocation.cells(2, _
105             -sensitivityInputs.Count - sensitivityOutputs.Count + 1)
106     Else
107         Call RunScenariosOnInputColumn(inputColIdx + 1)
108     End If
109     i = i + increment
110 Next
111 End Sub

```

Appendix C. Reusable Packaging CBA Source Code

This appendix complements the reusable packaging repair case study we present in Chapter 7.

C.1. Discrete Event Simulator

We implement the simulator as a VBA class. Each instance of the simulator tracks one type of material, such as boxes or box ties. The simulator tracks the age and location of each piece of material to determine when each piece is reused, discarded, or lost. While designed to work at week-level granularity over the course of three years, it may be easily augmented to support alternative pacing or duration.

In each period of the simulation (the simulation “event”), we invoke the following three methods:

1. **OrderTo()**: Adds needed material to inventory
2. **SendExchanges()**: Sends out material to customers
3. **ReceiveReturns()**: Receives material back from customers, less shrinkage and retired material, after the TAT has elapsed

We present the complete source code of this class in Listing 4.

Listing 4: Implementation of InventoryStore, a class to simulate the flow of shipping material over time

```
1
2 Class InventoryStore
3     Public OrderCount As Single
4     Public OrderQuantity As Single
5     Public TurnsPerUnit As Single
6     Public TotalInventory As Single
7     Public TurnaroundTime As Single
8     Public RetiredInventory As Single
9     Public InTransitInventory As Single
10    Public LostInventory As Single
11
12    Private Const MaxTurns As Integer = 90
13    Private Period As Integer
14
15    ' Tracks material in inventory. Index indicates number of uses remaining.
16    Private InventoryByUsesLeft(1 To MaxTurns) As Single
17
18    ' Tracks material out to customer. First index indicates period it will return.
19    ' Second index indicates number of uses remaining (including current usage).
20    Private ReturnsInPeriod(0 To 52 * 3 * 2, 1 To MaxTurns) As Single
21
22    Private Sub Class_Initialize()
23        OrderCount = 0
24        OrderQuantity = 1
25        TurnsPerUnit = 1
26        TotalInventory = 0
27        TurnaroundTime = 52 * 3 '3 years
28    End Sub
29
30    ' Ensures the desired amount of material is on hand. Orders more as necessary.
31    Public Sub OrderTo(ByVal targetInventory As Single)
32        If targetInventory > TotalInventory Then
33            Dim numOrders As Single
34            Dim newInventory As Single
35
```

```

36         numOrders = Application.WorksheetFunction.RoundUp( _
37             (targetInventory - TotalInventory) / OrderQuantity, 0)
38         newInventory = numOrders * OrderQuantity
39         TotalInventory = TotalInventory + newInventory
40         InventoryByUsesLeft(TurnsPerUnit) = InventoryByUsesLeft(TurnsPerUnit)
41             + newInventory
42         OrderCount = OrderCount + numOrders
43     End If
44 End Sub
45
46 ' Sends out the desired quantity of material to customers. Oldest material used first.
47 Public Sub SendExchanges(ByVal quantity As Single, ByVal shrinkagePct As Double)
48     TotalInventory = TotalInventory - quantity
49
50     Dim quantityReturning As Single
51     quantityReturning = quantity * (1 - shrinkagePct)
52
53     LostInventory = LostInventory + quantity - quantityReturning
54     InTransitInventory = InTransitInventory + quantityReturning
55
56     For i = 1 To MaxTurns
57         If quantity = 0 Then Exit For
58
59         Dim quantityWithThisUsageLeft As Single
60         quantityWithThisUsageLeft = InventoryByUsesLeft(i)
61
62         Dim quantityToSend As Single
63         Dim quantityToReturn As Single
64
65         If quantityWithThisUsageLeft > 0 Then
66             If (quantity >= quantityWithThisUsageLeft) Then
67                 quantityToSend = quantityWithThisUsageLeft
68                 quantity = quantity - quantityWithThisUsageLeft
69             Else
70                 quantityToSend = quantity
71                 quantity = 0
72             End If
73
74             If quantityReturning >= quantityToSend Then
75                 quantityToReturn = quantityToSend
76                 quantityReturning = quantityReturning - quantityToSend
77             Else
78                 quantityToReturn = quantityReturning
79                 quantityReturning = 0
80             End If
81
82             InventoryByUsesLeft(i) = InventoryByUsesLeft(i) - quantityToSend
83             ReturnsInPeriod(Period + TurnaroundTime, i) = _
84                 ReturnsInPeriod(Period + TurnaroundTime, i) + quantityToReturn
85         End If
86     Next
87 End Sub
88
89 ' Advances the simulated period by 1.
90 ' Recieves any material due back, less shrunk and retired inventory.
91 Public Sub ReceiveReturns()
92     Period = Period + 1
93     Dim totalReceived As Single
94
95     For i = 2 To MaxTurns
96         InventoryByUsesLeft(i - 1) = InventoryByUsesLeft(i - 1) _
97             + ReturnsInPeriod(Period, i)
98         totalReceived = totalReceived + ReturnsInPeriod(Period, i)

```

```

99         Next
100
101         TotalInventory = TotalInventory + totalReceived
102         RetiredInventory = RetiredInventory + ReturnsInPeriod(Period, 1)
103         InTransitInventory = InTransitInventory - totalReceived - ReturnsInPeriod(Period, 1)
104     End Sub
105 End Class

```

C.2. CBA Spreadsheet Integration

In Listing 5, we show how we integrate the InventoryStore class described in Section C.2 with Dell's CBA spreadsheet. For each of the four scenarios, the code reads inputs from the spreadsheet, runs two simulators (one for the box, another for the materials), and stores the output order quantities and timing in the spreadsheet.

Listing 5: Example usage of the simulator class

```

1
2 Sub CalcQuarters()
3     Dim sheet As Worksheet
4     sheet = Sheets.Item("Packaging Input")
5
6     ' Reset output cells in spreadsheet
7     For s = 0 To 3
8         For q = 1 To 4 * 3
9             sheet.Cells(33 + 4 * s, 2 + q).Value = 0
10            sheet.Cells(34 + 4 * s, 2 + q).Value = 0
11        Next q
12    Next s
13
14    ' Run each scenario
15    For s = 0 To 3
16        Dim materialPerExchange As Single
17        Dim shrinkagePct As Double
18        Dim inventoryBufferPct As Double
19
20        materialPerExchange = sheet.Cells(18, 3 + s).Value
21        shrinkagePct = sheet.Cells(20, 3 + s).Value
22        inventoryBufferPct = sheet.Cells(21, 3 + s).Value
23
24        ' Initialize box inventory
25        Dim boxInventory As InventoryStore : boxInventory = New InventoryStore
26        boxInventory.OrderQuantity = sheet.Cells(11, 3 + s).Value
27        boxInventory.TurnsPerUnit = sheet.Cells(13, 3 + s).Value
28        boxInventory.TurnaroundTime = sheet.Cells(22, 3 + s).Value
29
30        ' Initialize material inventory
31        Dim materialInventory As InventoryStore : materialInventory = New InventoryStore
32        materialInventory.OrderQuantity = sheet.Cells(16, 3 + s).Value
33
34        ' Run each quarter (for 3 years)
35        For q = 1 To 4 * 3
36            Dim demand As Single
37            Dim demandWithBuffer As Single
38
39            demand = sheet.Cells(27, 2 + q) / 13.0#

```

```

40      demandWithBuffer = demand * (1 + inventoryBufferPct)
41
42      ' Run each week in the quarter
43      For w = 1 To 13
44          ' Order
45          Call boxInventory.OrderTo(demandWithBuffer)
46          Call materialInventory.OrderTo(demandWithBuffer * materialPerExchange)
47
48          ' Send out material
49          Call boxInventory.SendExchanges(demand, shrinkagePct)
50          Call materialInventory.SendExchanges(demand * materialPerExchange, 0)
51
52          ' Receive returned material
53          Call boxInventory.ReceiveReturns()
54          Call materialInventory.ReceiveReturns()
55      Next w
56
57      ' Report counts in spreadsheet
58      sheet.Cells(33 + 4 * s, 2 + q).Value = boxInventory.OrderCount
59      sheet.Cells(34 + 4 * s, 2 + q).Value = materialInventory.OrderCount
60      sheet.Cells(53 + 5 * s, 2 + q).Value = boxInventory.TotalInventory _
61                                          + boxInventory.InTransitInventory
62      sheet.Cells(54 + 5 * s, 2 + q).Value = boxInventory.InTransitInventory
63      sheet.Cells(55 + 5 * s, 2 + q).Value = boxInventory.LostInventory
64      sheet.Cells(56 + 5 * s, 2 + q).Value = boxInventory.RetiredInventory
65
66      ' Reset counts in simulator
67      boxInventory.OrderCount = 0
68      materialInventory.OrderCount = 0
69      boxInventory.RetiredInventory = 0
70      boxInventory.LostInventory = 0
71      Next q
72      Next s
73      End Sub

```

Appendix D. Teardown Dashboard Source Code

This appendix supplements Chapter 8's teardown dashboard case study by providing code listings for the data architecture.

D.1. Connections Table

The Microsoft Access *Connections* table indicates data repository locations for the data to pull and summarize for use in the teardown dashboard. Each row, which defines one of the repository, also provides the structured query language (SQL) statement to run against the repository, the level of aggregation to apply (weekly, monthly, etc.), and other information as summarized in Table 54.

Table 54: Explanation of columns in the Connections table; this table identifies the data sources behind the teardown dashboard

Column name	Data type	Explanation
TableName	Text	The name of the Microsoft Access table where the aggregated data fetched from the data source should be stored locally
Enabled	Yes/No	Indicates whether data should be pulled from the data source
TargetStartDate	Date/Time	The start date and time of the data to be fetched (inclusive)
TargetEndDate	Date/Time	The end date and time of the data to be fetched (exclusive)
TargetGranularity	Text	The granularity to use when fetching data, which must be one of these values: <ul style="list-style-type: none">• Yearly• Quarterly• Monthly• Weekly• Daily
PurgeEndDate	Date/Time	The date before which data should be purged from the local table (exclusive)
ConnectionString	Text	The connection string that includes sufficient information to make a connection to the data source; this may include user credentials (a username and password), a database driver name, a database machine name or instance name, and/or a data source name (DSN)
SqlQuery	Memo	The SQL statement to run against the data source. The statement should contain placeholders for dates: <ul style="list-style-type: none">• %START_DATE% for the start of the range of data to aggregate (inclusive)• %END_DATE% for the end of the range of data to aggregate (exclusive)
DateColumnName	Text	The name of the column in the table identified by the TableName field where the timestamp of the data is stored; this is used to purge the data at the appropriate time.
LastUpdate	Date/Time	The last time data was fetched from this data source

D.2. Control Panel

The *Control Panel* window shown in Figure 67 is powered by the code in Listing 6. This code manipulates data in the Connections table to fetch data in either automatic or manual update mode. In automatic update mode, the code coordinates the aggregation of data in weekly increments over the past 5 weeks, summarizing and in monthly increments over the past 6 months; it also ensures data before the start of these ranges are purged.

Listing 6: Dashboard data control panel macros

```
1
2 Option Compare Database
3 Option Explicit On
4
5 Private updater As New TableUpdaterController
6
7 Private Sub cmdOK_Click()
8     If Not cmdOK.Enabled Then
9         If MsgBox("Are you sure you want to cancel processing?",
10             vbYesNo Or vbDefaultButton2 Or vbExclamation) = vbYes Then
11             updater.UpdateStatus("Aborted")
12             cmdOK.Enabled = True
13         End
14     End If
15 Else
16     cmdOK.Enabled = False
17     DoEvents()
18     If optAuto.Value Then
19         DoAutoUpdate()
20     Else
21         DoManualUpdate()
22     End If
23     DoEvents()
24     cmdCancel.Caption = "Close"
25     cmdOK.Visible = False
26 End If
27 End Sub
28
29 Private Sub Form_Load()
30     updater.StatusLine = lblStatus
31     updater.SubStatusLine = lblSubStatus
32     lblStatus.Caption = "Ready."
33     lblSubStatus.Caption = ""
34     optAuto.Value = True
35     optManual.Value = False
36     Connections.Visible = False
37 End Sub
38
39 Private Sub optAuto_AfterUpdate()
40     optManual.Value = False
41     Connections.Visible = False
42 End Sub
43
44 Private Sub optManual_AfterUpdate()
45     optAuto.Value = False
46     Connections.Visible = True
47 End Sub
48
49 Private Sub DoManualUpdate()
```

```

50     updater.NoUpdateIfDataExists = False
51     updater.UpdateAll()
52 End Sub
53
54 Private Sub DoAutoUpdate()
55     updater.NoUpdateIfDataExists = True
56
57     ' Do weekly updates
58     Dim startWeekly As Date, endWeekly As Date
59     endWeekly = RoundToInterval("ww", Date)
60     startWeekly = DateAdd("ww", -5, endWeekly)
61     CurrentDb.Execute("UPDATE Connections SET TargetStartDate=#" & startWeekly _
62         & "#, TargetEndDate=#" & endWeekly _
63         & "#, TargetGranularity='Weekly', PurgeEndDate=#" & startWeekly & "#")
64     updater.UpdateAll()
65
66     ' Do monthly updates
67     Dim startMonthly As Date, endMonthly As Date
68     endMonthly = RoundToInterval("m", Date)
69     startMonthly = DateAdd("m", -6, endMonthly)
70     CurrentDb.Execute("UPDATE Connections SET TargetStartDate=#" & startMonthly _
71         & "#, TargetEndDate=#" & endMonthly
72         & "#, TargetGranularity='Monthly', PurgeEndDate=#" & startMonthly & "#")
73     updater.UpdateAll()
74 End Sub

```

The `TableUpdaterController` class, defined in Listing 7, is an intermediary between the Connections table and the `TableUpdater` class, defined in Listing 8, which actually performs the data fetch and purge operations.

Listing 7: TableUpdaterController class

```

1
2 Class TableUpdaterController
3     Option Compare Database
4     Option Explicit
5
6     Public StatusLine As Label
7     Public SubStatusLine As Label
8     Public NoUpdateIfDataExists As Boolean
9
10    Public Sub UpdateStatus(msg As String)
11        StatusLine.Caption = Date & " " & Time & ": " & msg
12        DoEvents
13    End Sub
14
15    Public Sub UpdateAll()
16        Dim rsConn As Recordset
17        Set rsConn = CurrentDb("Connections").OpenRecordset()
18        Do While Not rsConn.EOF
19            If rsConn("Enabled") Then
20                UpdateTable rsConn
21                rsConn.Edit
22                rsConn("LastUpdate") = Now()
23                rsConn.Update
24                DoEvents
25            End If
26            rsConn.MoveNext
27        Loop

```

```

28     UpdateStatus "Finished updating data tables"
29 End Sub
30
31 Public Sub UpdateTable(r As Recordset)
32     UpdateStatus "Updating data table '" & r("TableName") & "'"
33
34     ' Initialize TableUpdater object
35     Dim updater As New TableUpdater
36     updater.TableName = r("TableName").Value
37     Set updater.StatusLine = SubStatusLine
38     updater.Dsn = r("ConnectionString").Value
39     updater.Sql = r("SqlQuery").Value
40     updater.DateColumnName = r("DateColumnName").Value
41     updater.IntervalColumnName = "Interval"
42     updater.Interval = r("TargetGranularity")
43
44     ' Purge old data
45     updater.PurgeOldData r("PurgeEndDate").Value
46
47     ' Perform the updates
48     Dim dateInterval As String
49     Dim startDate As Date, endDate As Date, nextDate As Date
50     Dim doUpdate As Boolean
51     dateInterval = GetDateIntervalByName(updater.Interval)
52     startDate = RoundToInterval(dateInterval, r("TargetStartDate").Value)
53     endDate = RoundToInterval(dateInterval, r("TargetEndDate").Value)
54     Do While startDate < endDate
55         ' Determine next interval
56         nextDate = DateAdd(dateInterval, 1, startDate)
57         updater.TargetStartDate = startDate
58         updater.TargetEndDate = nextDate
59
60         ' Check to see whether we should do an update for this interval
61         If NoUpdateIfDataExists Then
62             doUpdate = Not updater.DoesDataExist()
63         Else
64             doUpdate = True
65         End If
66
67         ' Do the update, if necessary
68         If doUpdate Then
69             updater.Update
70         End If
71
72         startDate = nextDate
73     Loop
74 End Sub
75 End Class

```

Listing 8: TableUpdater class

```

1
2 Class TableUpdater
3     Option Compare Database
4     Option Explicit
5
6     Public TableName As String
7     Public StatusLine As Label
8     Public Dsn As String
9     Public Sql As String
10    Public DateColumnName As String
11    Public IntervalColumnName As String
12    Public Interval As String
13    Public TargetStartDate As Date
14    Public TargetEndDate As Date
15
16    Private Sub Class_Initialize()
17        DateColumnName = "Month"
18    End Sub
19
20    Public Sub UpdateStatus(msg As String)
21        StatusLine.Caption = Date & " " & Time & ": " & msg
22        DoEvents
23    End Sub
24
25    ' Perform both data purgin and loading activities
26    Public Sub Update()
27        ' Purge data within load range
28        UpdateStatus "Purging data from " & TargetStartDate & " to " & TargetEndDate _
29            & IIf(Interval <> "", " (" & Interval & ")", "")
30        PurgeData TargetStartDate, TargetEndDate
31
32        ' Load appropriate data
33        Dim rowCount As Integer
34        UpdateStatus "Fetching data from " & TargetStartDate & " to " & TargetEndDate _
35            & IIf(Interval <> "", " (" & Interval & ")", "")
36        rowCount = LoadData(TargetStartDate, TargetEndDate)
37
38        ' Refresh related pivots
39        UpdateStatus "Load complete with " & rowCount & " row(s)"
40    End Sub
41
42    ' Determines whether any data before the target start date exists with
43    ' aggregated by the target aggregation unit (quarter, month, etc.)
44    Public Function DoesDataExist() As Boolean
45        Dim query As String
46        query = "SELECT COUNT(*) FROM " & TableName & " WHERE [" & DateColumnName _
47            & "] >= #" & TargetStartDate & " AND [" & DateColumnName & "] < #" _
48            & TargetEndDate & "#"
49        If IntervalColumnName <> "" Then
50            query = query & " AND [" & IntervalColumnName & "] = '" & Interval & "'"
51        End If
52
53        Dim rs As Recordset
54        Set rs = CurrentDb.OpenRecordset(query)
55        DoesDataExist = rs(0).Value > 0
56    End Function
57
58    ' Purges all data before the particular end date (exclusive) with the target interval
59    Public Sub PurgeOldData(PurgeEndDate As Date)
60        UpdateStatus "Purging data before " & PurgeEndDate
61        PurgeData #1/1/1900#, PurgeEndDate
62    End Sub

```

```

63
64 ' Purges all data within the start (inclusive) and end (exclusive) range with
65 ' the target interval
66 Private Sub PurgeData(startDate As Date, endDate As Date)
67     Dim query As String
68     query = "DELETE FROM " & TableName & " WHERE [" & DateColumnName & "] >= #" & _
69         startDate & "# AND [" & DateColumnName & "] < #" & endDate & "#"
70     If IntervalColumnName <> "" Then
71         query = query & " AND [" & IntervalColumnName & "] = '" & Interval & "'"
72     End If
73     CurrentDb.Execute query
74 End Sub
75
76 ' Loads data within a given start (include) and end (exclusive) range
77 Private Function LoadData(startDate As Date, endDate As Date) As Integer
78     ' Prepare query
79     Dim query As String: query = sql
80     query = Replace(Replace(query, "%START_DATE%", "" & startDate & ""), _
81         "%END_DATE%", "" & endDate & "")
82
83     ' Open output recordset
84     Dim rsOutput As Recordset
85     Set rsOutput = CurrentDb.OpenRecordset(TableName)
86
87     ' Open input recordset
88     Dim db As New ADODB.Connection
89     Dim rsInput As ADODB.Recordset
90     db.Open Dsn
91     db.CommandTimeout = 60 * 60
92     Set rsInput = db.Execute(query)
93
94     ' Insert data into table
95     Dim f As ADODB.Field
96     Dim fieldIdx As Integer
97     Dim rowCount As Integer
98     Dim fieldIndices() As Integer
99     Dim intervalIdx As Integer
100     Do While Not rsInput.EOF
101         rowCount = rowCount + 1
102         If rowCount = 1 Or rowCount Mod 50 = 0 Then
103             UpdateStatus "Adding row " & rowCount
104         End If
105
106         ' First time through, grab the ordinal
107         If rowCount = 1 Then
108             ReDim fieldIndices(rsInput.Fields.Count)
109             fieldIdx = 0
110             For Each f In rsInput.Fields
111                 fieldIndices(fieldIdx) = rsOutput.Fields(f.name).OrdinalPosition
112                 fieldIdx = fieldIdx + 1
113             Next
114             If IntervalColumnName <> "" Then
115                 intervalIdx = rsOutput.Fields(IntervalColumnName).OrdinalPosition
116             Else
117                 intervalIdx = -1
118             End If
119         End If
120
121         ' Copy the data from the input recordset to the output recordset
122         rsOutput.AddNew
123         For fieldIdx = rsInput.Fields.Count - 1 To 0 Step -1
124             rsOutput.Fields(fieldIndices(fieldIdx)).Value = _
125                 rsInput.Fields(fieldIdx).Value

```



```
126         Next
127         If intervalIdx >= 0 Then
128             rsOutput(intervalIdx) = Interval
129         End If
130         rsOutput.Update
131
132         rsInput.MoveNext
133     Loop
134     LoadData = rowCount
135 End Function
136 End Class
```

References

- [1] Jason Snell. (2010, December) Macworld. [Online].
<http://www.macworld.com/article/156506/2010/12/applefin.html>
- [2] Aberdeen Group, "Industry Best Practices in Reverse Logistics," January 2007.
- [3] Bastiaan Janse, Peter Schuur, and Marisa P. de Brito, "A reverse logistics diagnostic tool: the case of the consumer electronics industry," 2009.
- [4] Joakim Wikner and Ou Tang, "A structural framework for closed-loop supply chains," 2008.
- [5] Robert Scanlon, "Aligning Product and Supply Chain Strategies in the Mobile Phone Industry," 2009.
- [6] J.R. Stock, *Reverse Logistics*. Oak Brook, IL, 1992.
- [7] Dale S. Rogers and Ronald S. Tibben-Lembke, "Going Backwards: Reverse Logistics Trends and Practices," 1999.
- [8] V. Daniel R. Guide Jr. and Luk N. Van Wassenhove, "The Evolution of Closed-Loop Supply Chain Research," 2008.
- [9] James Stock, Thomas Speh, and Herbert Shear, "Many (Happy) Product Returns," *Harvard Business Review*, vol. 80, no. 7, pp. 16-17, July 2002.
- [10] Moritz Fleischmann, Jo A. E. E. van Nunen, and Ben Grave, "Integrating Closed-Loop Supply and Spare-Parts Management at IBM," 2003.
- [11] IBM. (2010) 40 Years of IBM Environment Leadership. [Online].
http://www.ibm.com/ibm/environment/information/40_Years_of_IBM_Environmental_Leadership.pdf
- [12] B Willems, W. Dewulf, and J. R. Duflou, "Can large-scale disassembly be profitable? A linear programming approach to quantifying the turning point to make disassembly economically viable," 2005.
- [13] Sameer Kumar and Valora Putnam, "Cradle to cradle: Reverse logistics strategies and opportunities across three industry sectors," *International Journal of Production Economics*, no. 115, pp. 305-315, 2008.
- [14] Patricia J. Daugherty, R. Glenn Richey, Stefan E. Genchev, and Haozhe Chen, "Reverse logistics: superior performance through focused resource commitments to information technology," 2004.
- [15] Shad Dowlatshahi, "The role of transportation in the design and implementation of reverse logistics systems," 2010.
- [16] H. Hamza, Y. Wang, and B. Bidanda, "Modeling total cost of ownership utilizing interval-based reliable simulation technique in reverse logistics management," 2007.
- [17] Moritz Fleischmann et al., "Quantitative models for reverse logistics: A review," 1997.
- [18] Dale S. Rogers and Ronald Tibben-Lembke, "An Examination of Reverse Logistics Practices," 2001.
- [19] Felix T.S. Chan and Hing Kai Chan, "A survey on reverse logistics system of mobile phone industry in Hong Kong," 2008.
- [20] Stefan E. Genchev, "Reverse logistics program design: A company study," 2009.
- [21] Ramesh Subramoniam, Donald Huisinigh, and Ratna Babu Chinnam, "Aftermarket remanufacturing strategic planning decision-making framework: theory & practice," vol. 18, no. 16-17, November 2010.
- [22] Thomas Spengler and Marcus Shroter, "Strategic Management of Spare Parts in Closed-Loop Supply Chains -- A System Dynamics Approach," 2003.
- [23] Albert Wee Kwan Tan and Arun Kumar, "A decision-making model for reverse logistics in the computer industry," 2006.

- [24] Patroklos Georgiadis, Dimitrios Vlachos, and George Tagaras, "The Impact of Product Lifecycle on Capacity Planning of Closed-Loop Supply Chains with Remanufacturing," 2006.
- [25] V. Daniel Guide Jr., Gilvan C. Souza, Luk N. Van Wassenhove, and Joseph D. Blackburn, "Time Value of Commercial Product Returns," 2006.
- [26] Joseph D. Blackburn, Danie R. Guide Jr., Gilvan C. Souza, and Luk N Van Wassenhove, "Reverse Supply Chains for Commercial Returns," *California Management Review*, vol. 46, no. 2, pp. 6-22. 2004.
- [27] Jessica Hanafi and Sami Kara and Hartmut Kaebnick, "Reverse logistics strategies for end-of-life products," 2008.
- [28] Akihiro Kaga, "Application of Real Options to Reverse Logistics Process," *MIT Thesis*, 2004.
- [29] Christopher C Lin, "A Decision System for Routing Returned Product to the Optimal Recovery Channel," *MIT Thesis*, 2010.
- [30] Eduard Shevtshenko, "Decision support under uncertainties based on robust Bayesian networks in reverse logistics management," 2009.
- [31] E Akçali, S. Çetinkaya, and H. Üster, "Network Design for Reverse and Closed-Loop Supply Chains: An Annotated Bibliography of Models and Solution Approaches," 2009.
- [32] Moritz Fleischmann, Patrick Beullens, Jacqueline M. Bloemhof-Ruwaard, and Luk N. van Wassenhove, "The Impact of Product Recovery on Logistics Network Design," 2000.
- [33] MI Salema, APB Póvoa, and AQ Novais, "A warehouse-based design model for reverse logistics," 2006.
- [34] V. Daniel Guide Jr., Luc Muyldermans, and Luk N. Van Wassenhove, "Hewlett-Packard Company Unlocks the Value Potential from Time-Sensitive Returns," 2005.
- [35] Vaidyanathan Jayaraman, "Production planning for closed-loop supply chains with product recovery and reuse: an analytical approach," 2005.
- [36] Kris Lieckens and Nico Vandaele, "Reverse logistics network design with stochastic lead times," 2005.
- [37] Hokey Min, Chang-Seong Ko, and Hyun-Jeung Ko, "The spatial and temporal consolidation of returned products in a closed-loop supply chain network," 2006.
- [38] Hokey Min and Hyun-Jeung Ko, "The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers," 2007.
- [39] W.-H. Tsai and Shih-Jieh Hung, "A fuzzy goal programming approach for green supply chain optimisation under activity-based costing and performance evaluation with a value-chain structure," 2007.
- [40] Mir Saman Pishvaei, Kamran Kianfar, and Behrooz Karimi, "Reverse logistics network design using simulated annealing," 2008.
- [41] Maria Isabel Gomes Salema, Ana Paula Barbosa-Póvoa, and Augusto Q. Novais, "Simultaneous design and planning of supply chains with reverse flows: A generic modelling framework," 2008.
- [42] Der-Hong Lee and Meng Dong, "A heuristic approach to logistics network design for end-of-lease computer products recovery," 2006.
- [43] V. Daniel R. Guide Jr. and Luk N. Van Wassenhove, "Closed-Loop Supply Chains: An Introduction to the Feature Issue: Parts 1 & 2," 2006.
- [44] Deborah Ancona, Thomas A. Kochan, Maureen Scully, John Van Maanen, and D. Eleanor Westney, *Managing for the Future: Organizational Behavior & Processes*.: South-Western College Publishers, 2005.
- [45] Michael Dell. (2010, June) Dell - The Official Site. [Online]. <http://content.dell.com/us/en/corp/d/corp-comm/fy10-year-in-review.aspx>
- [46] Dell Inc., "Form 10-K," Filing to the U.S. SEC 2010.
- [47] Peter Judge, "Dell's CIO: Virtualisation Saved Us £100 Million," *eWeek Europe*, July 2010.

- [48] Grace E. Kerry, "HP Ends IT Restructuring Process," December 2008.
- [49] Charles H. Fine, *Clockspeed*. Reading, MA: Perseus Books, 1998.
- [50] R. Canan Savaskan, Shantanu Bhattacharya, and Luk N. Van Wassenhove, "Closed-Loop Supply Chain Models with Product Remanufacturing," *Management Science*, vol. 50, no. 2, pp. 239-252, February 2004.
- [51] David Segrera, "Finished Goods Inventory Strategy: Positioning and Leveling Analysis," *MIT Thesis*, June 2011.
- [52] Jakob Krarup and Peter Mark Pruzan, "The simple plant location problem: Survey and synthesis," *European Journal of Operational Research*, vol. 36, no. 81, pp. 36-81, May 1983.
- [53] Roland Geyer and Vered Doctori Blass, "The economics of cell phone reuse and recycling," 2010.
- [54] Marc Gunther. (2007, March) Fortune. [Online].
http://money.cnn.com/2007/03/08/magazines/fortune/pluggedin_gunther_dellrecycle.fortune/index.htm
- [55] Use Reusables. Reusables Cost Comparison Tool. [Online].
<http://www.usereusables.com/cost/cctool.html>
- [56] M. Wattenberg, "Visualizing the stock market," *Proceedings of Extended Abstracts of Human Factors in Computing Systems*, pp. 188-189, 1999.
- [57] M. Bruls, K. Huizing, and J.J. Wijk, "Squarified treemaps," *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization*, pp. 33-42, 2000.
- [58] Guide Jr. and Daniel R., "The Development of Closed-Loop Supply Chains," *CIMSO Supply Chain Forum*, March 2005.
- [59] Ron Giuntini and Kevin Gaudette, "Remanufacturing: The next great opportunity for boosting US productivity," *Business Horizons*, pp. 41-48, November-December 2003.
- [60] Jean-Baptiste Michel and et al., "Quantitative Analysis of Culture Using Millions of Digitized Books," *Science*, vol. 331, no. 6014, pp. 176-182, January 2011.
- [61] Gailen Vick, "Directing Reverse Logistics -- A Corporate Paradigm Shift," *Reverse Logistics Digital Magazine*, no. 23, pp. 26-28, May 2010.
- [62] Automotive Aftermarket Industry Association. [Online]. <http://www.aftermarket.org>
- [63] Reverse Logistics Association. [Online]. <http://www.reverselogisticstrends.com>